Stochastic Nested Aggregation for Images and Random Fields

by

Slawomir Bogumil Wesolkowski

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2007

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Slawomir Bogumil Wesolkowski

Abstract

Image segmentation is a critical step in building a computer vision algorithm that is able to distinguish between separate objects in an image scene. Image segmentation is based on two fundamentally intertwined components: pixel comparison and pixel grouping. In the pixel comparison step, pixels are determined to be similar or different from each other. In pixel grouping, those pixels which are similar are grouped together to form meaningful regions which can later be processed. This thesis makes original contributions to both of those areas.

First, given a Markov Random Field framework, a Stochastic Nested Aggregation (SNA) framework for pixel and region grouping is presented and thoroughly analyzed using a Potts model. This framework is applicable in general to graph partitioning and discrete estimation problems where pairwise energy models are used. Nested aggregation reduces the computational complexity of stochastic algorithms such as Simulated Annealing to order $O(N)$ while at the same time allowing local deterministic approaches such as Iterated Conditional Modes to escape most local minima in order to become a global deterministic optimization method. SNA is further enhanced by the introduction of a Graduated Models strategy which allows an optimization algorithm to converge to the model via several intermediary models. A well-known special case of Graduated Models is the Highest Confidence First algorithm which merges pixels or regions that give the highest global energy decrease. Finally, SNA allows us to use different models at different levels of coarseness. For coarser levels, a mean-based Potts model is introduced in order to compute region-to-region gradients based on the region mean and not edge gradients.

Second, we develop a probabilistic framework based on hypothesis testing in order to achieve color constancy in image segmentation. We develop three new shading invariant semi-metrics based on the Dichromatic Reflection Model. An RGB image is transformed into an $R'G'B'$ highlight invariant space to remove any highlight components, and only the component representing color hue is preserved to remove shading effects. This transformation is applied successfully to one of the proposed distance measures. The probabilistic semi-metrics show similar performance to vector angle on images without saturated highlight pixels; however, for saturated regions, as well as very low intensity pixels, the probabilistic distance measures outperform vector angle.

Third, for interferometric Synthetic Aperture Radar image processing we apply the Potts model using SNA to the phase unwrapping problem. We devise a new distance measure for identifying phase discontinuities based on the minimum coherence of two adjacent pixels and their phase difference. As a comparison we use the probabilistic cost function of Carballo [16] as a distance measure for our experiments.

Acknowledgements

I am indebted to my supervisor, Dr. Paul Fieguth, for motivating me and keeping me focused on what was important. Without his expert guidance, this thesis would have never seen the light. I would also like to thank the members of my examining committee (Dr. Prabir Bhattacharya, Dr. George Freeman, Dr. Mohammed Kamel, and Dr. Stephen Murphy) for their careful reading of the thesis and suggestions for improvements.

Last but not least, I would like to thank my wife Anna for her love and always believing this endeavor would succeed and my parents for their constant support for all my enterprizes.

To Anna and Emilia

Contents

List of Tables

List of Figures

List of Algorithms

Nomenclature

Basic Definitions

Syntax	Definition
a	scalar, random variable
$\underline{a}$	column vector, random vector
a_i	ith component of vector $\underline{a}$
$\underline{a}_i$	ith vector in a sequence
A	matrix
$a_{i,j}$	(i,j)th element of matrix A
A^T	matrix transpose
A^{-1}	matrix inverse
$\underline{\bar{a}}$	shading invariant $\underline{a}$
$\underline{\tilde{a}}$	highlight invariant $\underline{a}$
$\underline{\bar{\tilde{a}}}$	highlight and shading invariant $\underline{a}$
d	number of elements in a vector or its dimension

Graph Partition and Image Segmentation

Syntax	Definition
$\mathcal{S}$	lattice of sites $\{(i,j)\}$ or $\{i\}$ representing pixel locations in an image
$(i,j), i$	site in lattice $\mathcal{S}$ (will be used interchangeably)
w	number of columns in a regular lattice
h	number of rows in a regular lattice
N	size of lattice $\mathcal{S}$ with $N = w \cdot h$ for a regular lattice.
X	image pixel values on $\mathcal{S}$ with width w and height h.
$\underline{x}_i$, $\underline{x}_v$	characteristics of or features associated with vertex/node v or site i
$\mathcal{G}$	adjacency graph such that $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$.

Syntax	Definition
$\mathcal{V}$	set of nodes that need to be partitioned where $\mathcal{V} = \{v_1, v_2, ..., v_N\}$
v, v_i, i	vertex of graph $\mathcal{G}$
$\mathcal{E}$	the set of edges connecting adjacent nodes where $\mathcal{E} = \{(v_i, v_j)\}$
V_i	subset of nodes i or v_i in set $\mathcal{V}$
ℓ	set of random label variables (the random field)
$\mathcal{L}$	set of labels to be assigned such that $\mathcal{L} \in \{1, 2, \ldots, K\}$
K	total number of labels used
k	a label in set $\mathcal{L}$
$l_{i,j}$, l_i	label for pixel (i, j) or node i from the set of all possible labels in $\mathcal{L}$
$\mathcal{N}$	neighborhood system
$\mathcal{N}_i$	neighborhood system for pixel or node i
π_n	n-partition of graph $\mathcal{G}$
n	number of partitions in graph $\mathcal{G}$
W	world representation
Ω, Ω_π	solution space for W
Ω_{π_n}	space of all possible n-partitions π_n of $\mathcal{V}$
W^*	solution
l_{V_i}	label of subset V_i
s	level on bottom-up hierarchy, where $s = 0$ corresponds to the finest (or pixel) level
$\mathcal{G}^{(s)}$	adjacency graph at level s
$l_i^{(s)}$	label for node i at level s

Syntax	Definition
$n_i^{(s)}$	number of pixels in region i at level s
U	global energy
$U^{(s)}$	global energy at level s
$U_i^{(s)}$	energy of node i at level s
$\mathcal{R}$	set of all nodes i that form a region
$\mathcal{R}^{(s)}$	set of all nodes i that form a region at level s
$G_i^{(s)}$	nodes in $\mathcal{R}^{(s-1)}$ contained in i
$\Phi_{i,j}$	dissimilarity criterion or distance measure between nodes i and j; equivalent to $\Phi(i,j)$
$\Phi_{i,j}^{(s)}$	distance measure between regions i and j at level s
$\beta_{i,j}$	region coupling parameter between regions i and j; equivalent to $\beta(i,j)$
$\beta_{i,j}^{(s)}$	region coupling parameter between regions i and j at level s
$\delta_{i,j}$	Kronecker delta between vertices v_i and v_j (equivalently between sites i and j)
$\underline{w}_k$	prototype vector for class k or region with label k
$\mathcal{S}_i$	lattice i for Besag's "coding" method such that $\mathcal{S} = \cup_i \mathcal{S}_i$

Phase Unwrapping

Syntax	Definition
P_i	measured phase at pixel i
C_i	measured coherence at pixel i
$\Phi_m(i,j)$	distance between two phases based on phase difference and minimum coherence

Color Science and Color Distance Measures

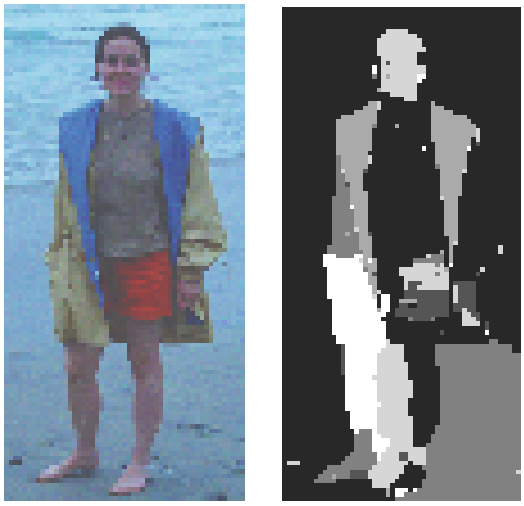
Syntax	Definition
R	red filter sensor response for a camera
G	green filter sensor response for a camera
B	blue filter sensor response for a camera
o	object surface
λ	wavelength of reflected light
$c^o(\lambda, i, j)$	color signal: light reflected from an object surface o
$e(\lambda)$	spectral power distribution of a light source
$s^o(\lambda)$	spectral-surface reflectance of an object o
$\nu_{i,j}$	shading factor
$\eta_{i,j}$	scalar specular reflection factor
$\upsilon_{i,j}$	residual noise
$\mathcal{R}_i(\lambda)$	spectral sensitivity functions of the camera in the visible spectrum where $i \in \{R, G, B\}$
$\underline{c_b}$	body color vector
$\underline{c_i}$	illumination color vector (specular reflection)
H	highlight invariant linear transformation
R_x	the noise covariance matrix for pixel $\underline{x}$
$\Phi_E(i, j)$	Euclidean distance measure
$\Phi_V(i, j)$	vector angle distance measure
$\Phi_{VH}(i, j)$	vector angle distance measure for highlight invariant color space
$\Phi_M(i, j)$	Mahalanobis distance measure
$\Phi_S(i, j)$	Same mean probabilistic distance measure
$\Phi_C(i, j)$	Common mean probabilistic distance measure

Chapter 1

Introduction

Humans can easily, even effortlessly, distinguish between separate objects in an image scene. This has long been a key problem in computer vision where a number of steps, from low-level to high-level vision, are needed to understand an image or some portion of it. A critical step is that of image segmentation [58, 105] — the partitioning of an image into distinguishable subsets based on the premise that objects having a distinct appearance can be visually separated. Images are composed of pixels which, depending on the sensors used to capture them, can represent light intensity values, colors or some other electromagnetic quantities. Image segmentation requires two distinct components: pixel comparison and pixel grouping. The pixel comparison function requires the design of a pixel similarity criterion. The pixel grouping mechanism, on the other hand, aggregates the pixels with respect to this pixel similarity criterion.

In this thesis, the similarity criterion is examined in light of the advances in the fields of color image segmentation [5, 24, 93], as well as in phase unwrapping [117]. In color image segmentation, image partitioning is carried out using chromatic information which provides a rich set of object cues over and above the brightness levels or textures available in grayscale images. The chromatic information can improve performance in a variety of applications such as video surveillance, face recognition, medical imaging, image retrieval from the Internet or specialized libraries, or color map segmentation. Figure 1.1 shows a color image and its segmentation. One can already appreciate important challenges such as region spilling (notice how one of the legs merges with part of the background).



(a) (b)

Figure 1.1: An example color image segmentation: (a) original image, (b) image segmentation.

While scene analysis using image segmentation has been widely reported in the literature, the use of image segmentation for phase unwrapping has been less frequent [117, 162]. Phase unwrapping allows us to examine Synthetic Aperture Radar (SAR) interferometry problems where phase and coherence information can be used to create topographic maps. Figure 1.2 shows images corresponding to the phase and coherence obtained over Mt. Vesuvius, as well as a typical segmentation result and the digital elevation model. In this application, similar challenges to color image segmentation exist. Region spilling is a serious problem which can have global effects (e.g., wrong elevation being estimated as is the case in Figure 1.2).

The pixel comparison and pixel grouping components can be both encoded in an energy minimization framework derived from stochastic physics called Markov Random Fields (MRFs) [22, 48, 88, 157]. Briefly, MRFs allow us to solve image segmentation problems using contextual constraints with respect to a chosen pixel comparison criterion. MRF modelling is an optimization framework that can be either deterministic [7](greedy methods such as gradient descent which always choose the lowest energy) or stochastic [48] (does not always choose a lower energy formulation allowing escapes from local minima). Each of these alternative paradigms has its own advantages and disadvantages. In the next two sections, pixel comparison and pixel grouping will be examined further and thesis contributions will be outlined in Section 1.3.

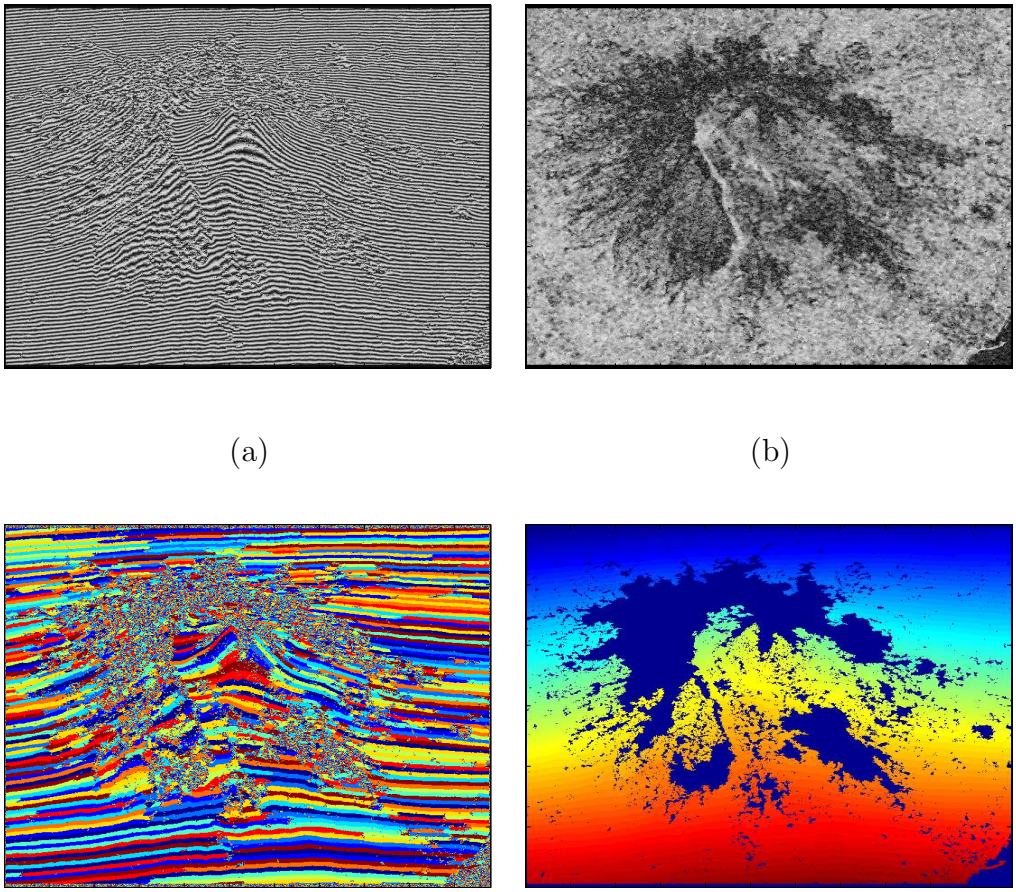
1.1 Pixel Comparison

In pixel comparison, a between-pixel similarity criterion or measure is needed. This criterion needs to reflect the kind of problem that is being solved. In other words, knowledge about the problem is encoded in the distance measure. Suppose we have two pixels $\underline{x}$ and $\underline{y}$. The distance (in some feature space) between these two points can be characterized by a pairwise distance

$$\Phi_g(\underline{x}, \underline{y}) = g(\underline{x}, \underline{y}) \tag{1.1}$$

where g is a function of the two pixels.

There are two general ways how this can be accomplished. First, one can extract features from the image pixels, effectively transforming the image originally in the sensor



(a) (b)

(c) (d)

Figure 1.2: An example of phase unwrapping for creating digital elevation models (DEMs) using image segmentation: (a) original phase image, (b) original coherence image (each pixel indicates the reliability of the corresponding phase pixel), (c) segmented phase image, and (d) digital elevation model (blue is the highest elevation while red the lowest). Notice the various discontinuities in (d).

space into anther space [35]. We can assume that a feature space is isotropic, thereby, computing similarities using the Euclidean distance

$$\Phi_E(\underline{x}, \underline{y}) = (\underline{x} - \underline{y})^T(\underline{x} - \underline{y}). \tag{1.2}$$

However, the isotropic assumption is not always valid. Therefore, the alternative is to build the feature comparison directly into the pixel distance measure, thereby, operating directly on the original data [35]. One example of a distance measure which operates directly on the sensor space is the vector angle [127, 148]

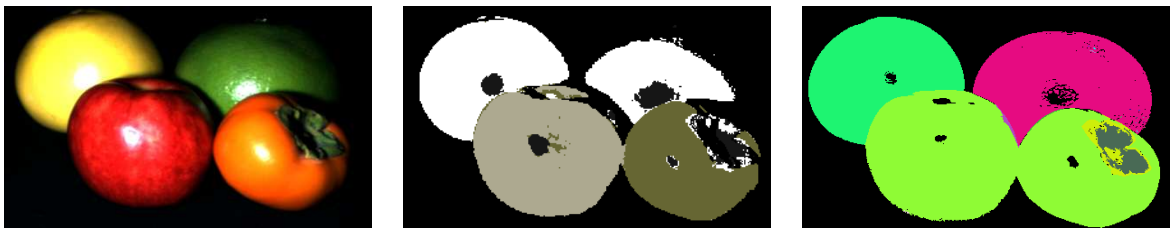
$$\Phi_V(\underline{x}, \underline{y}) = 1 - \frac{\underline{x}^T \underline{y}}{|\underline{x}| \cdot |\underline{y}|} \tag{1.3}$$

where $|\cdot|$ defines the L^2 norm. By working directly on the raw data, the vector angle avoids assuming that the data space is isotropic and is able to compare color pixels irrespective of the illumination intensity. This allows for the design of appropriate similarity measures that can operate without intermediary features, which are usually complex and sometimes computationally expensive to obtain. However, in the specific case of vector angle, noise statistics are not preserved due to the non-linear transformation. In addition, distances for pixels with low intensity values are ill-defined [127, 149].

The pixel comparison measures presented in this thesis will, therefore, be derived from basic principles to work directly on the sensor space (*i.e.*, either color images or the phase unwrapping problem). We will endeavor to present metrics which are noise invariant and carry out reflectance-based distance computation.

1.2 Pixel Grouping

Once pixels can be meaningfully compared to each other, a grouping mechanism needs to be found to enable regions to be formed and images to be analyzed. In general, grouping algorithms can be considered visual labelling, graph partition, graph coloring, or discrete state estimation methods (all terms will be treated as equivalent in this thesis). The goal is to assign a set of labels $\mathcal{L} = \{1, 2, \ldots, K\}$ to a set of pixels $\{\underline{x}_i\}$ on a regular lattice or grid $\mathcal{S} = \{i\}$ of size N. This results in the configuration or solution space $\Omega = \mathcal{L}^N$. If the



(a) (b) (c)

Figure 1.3: Comparing spatial aggregation and feature-based clustering for image segmentation: (a) original image, (b) image segmented using a feature-based clustering method [150], and (c) image segmented using a spatial aggregation method [151]. Notice the well formed regions in (b) with two regions merged in feature space whereas (c) contains some region-to-region spilling since the two regions contain a smooth transition point at their border.

label set is discrete with K labels, the size of the resulting space is K^N which for images is very large; for a continuous label set, this space is infinite.

A graph and its constituent nodes correspond to an image and its pixels respectively[1]. Image segmentation methods can be roughly categorized as follows: spatial aggregation, and feature-based clustering. Spatial aggregation concerns all methods that use spatial relationships to build regions of interest [58] whereas feature-based clustering methods aggregate pixels together based on their similarity into groups or clusters and usually do not use spatial information [35] (in some cases spatial information is used [64, 165]). In clustering-based approaches, there is usually no guarantee of spatial compactness, whereas spatially-based methods suffer from region-to-region spilling (which may occur when two different color regions, connected by a slowly changing gradient, are joined together). Figure 1.3 illustrates the difference between the two paradigms of image segmentation[2].

Most algorithms presented in the literature are ad-hoc. However, one way to transition to a principled framework involves finding the optimal solution to an energy function or

[1]Graphs will be formally defined in the next chapter

[2]Clustering results for the fruit image originally published in [150].

model encompassing spatial aggregation, feature-based clustering or both. The optimal solution to fitting the model to the data can be obtained using local (usually deterministic) or global (usually stochastic) optimization methods. Using spatial aggregation-type methods based on an energy framework enables us to design an energy function based on a *pairwise* comparison predicate between *adjacent* pixels. In clustering, since spatial information is usually not used, pixels that are not adjacent can be grouped together based only on the distance measure.

A significant enhancement of these energy-based methods comes in the form of contextual constraints. Contextual constraints can be very easily encoded by using Markov Random Fields (MRFs) [48, 88, 157] and can be applied to both spatial aggregation and feature-based clustering methods. In the latter case, clustering algorithms are able to integrate spatial information with clusters in the feature space [107]. MRFs allow us to conditionally decorrelate the assignment of a label to a pixel from its neighbors. In other words,

$$p(l_i|l_{\mathcal{N}_i}) = p(l_i| \cup_{\forall j \neq i} l_j) \tag{1.4}$$

where l_i is the label and $\mathcal{N}_i$ is the local neighborhood system of pixel i. There are also other energy based methods which are not based on MRFs and which will be reviewed in the thesis [167].

In general, energy-based pixel grouping mechanisms [22, 48, 88, 157, 167] rely on the use of sampling methods. This thesis focuses on Markov Chain Monte Carlo (MCMC) sampling methods such as Gibbs sampling. These methods are at a considerable disadvantage due to their high computational complexity when trying to obtain a global minimum. Since images are usually large, the sampling task becomes impractical as the computational complexity increases quickly with the number of pixels that need to be processed.

Stochastic and deterministic optimization algorithms are needed for MCMC since the solution space Ω is very large. A popular stochastic optimization algorithm is simulated annealing (SA) [19, 48, 76]. Simulated annealing allows solutions to increase in energy with non-zero probability, thereby, allowing SA to escape local minima in order to reach the global optimum. However, SA is usually run with sub-optimal parameters in order to speed up its convergence. Optimal parameters lead effectively to an exhaustive search of

Ω since the search needs to be carried out for an infinite number of iterations [48]. This is not an adequate conclusion for practical problems.

Therefore, sub-optimal parameters lead to a lower likelihood of escaping a particularly deep local minimum and, therefore, the uncertainty of reaching the global optimum. In the limit, when SA has zero probability of increasing the energy, it becomes a deterministic method where escaping from a local minimum is impossible (cf. Iterated Conditional Modes in Section 2.7). Therefore, in practice optimization algorithms seek a local solution (which is hopefully close to the global optimum). Due to the computational complexity of stochastic optimization algorithms such as SA and problems with global convergence of local methods such as ICM, one needs a method to accelerate optimization to achieve the full benefit of Gibbs sampling.

1.3 Thesis Contributions

The main motivation of this thesis is to contribute to the state of the art in pixel similarity and pixel grouping methods.

The following contributions augment the state of the art in pixel grouping algorithms in particular and graph partitioning methods in general:

- We introduce Stochastic Nested Aggregation (SNA), a method which accelerates discrete state estimation or graph partitioning using stochastic or deterministic approaches through a hierarchy of graph partitions (in the case of image processing, it is a hierarchy of image segmentations) in order to minimize a single global criterion. Stochastic nested aggregation can significantly speed-up stochastic algorithms such as simulated annealing [48, 76] by allowing fast convergence of the Gibbs sampler to a stationary probability distribution of the label random field.

  There are two main differences between this method and other bottom-up pixel-aggregation methods. First, its purpose is to discover an optimal stationary probability corresponding to the optimal partitioning of the graph which usually contains more than one node (as opposed to merging all pixels/nodes into one region/node at the ultimate level [99] or applying a stopping criterion at a lower level). Second, it is scale invariant in that the local minimum of the first level of the hierarchy is

identical to that of the coarsest (highest level in the graph) level in the hierarchy. It's computational complexity is $O(N)$ which is a considerable improvement over $O(N^3)$ for standard Gibbs sampling. The speed-up is more significant when homogenous regions within an image are large thanks to the pyramidal structure of nested aggregation. In practical terms, Gibbs sampling can be sped up by a factor of 1000-10000 (or more) depending on the graph size and the size of the largest partition in the coarsest level graph. Furthermore, it is stochastic in nature at each level of the hierarchy and, therefore, moves are reversible within the level (but not between levels). Finally, it is restricted to energy models with pairwise comparisons such as the first order Potts model.

- SNA transforms local optimization methods such as Iterated Conditional Modes (ICM) [7] into global optimizers. First, as for stochastic methods, there is a considerable computational speed-up obtained due to the use of reduced order graphs at each level of the hierarchy thus reducing the number of nodes being processed as the irregular bottom-up pyramid grows. Second, the major problem of local deterministic methods is that they get stuck in local minima. Nested aggregation can break label configuration deadlocks which give rise to these local minima by creating a new reduced order graph that no longer contains the same node configuration thus breaking the deadlock. It is thus able to reach a good local minimum from a random label initialization.

- We present a Graduated Models strategy for Stochastic Nested Aggregation in order to avoid getting stuck in an undesirable local minimum (*e.g.*, avoid region-to-region spilling in image segmentation). We apply Graduated Models to the Potts energy model where we vary the region coupling parameter from a low value (all pixels or nodes are their own regions) to the desired value (where regions homogenous in features have formed). Therefore, through careful nested aggregation simulated annealing and ICM converge to a very good local minimum.

- SNA allows us to change models at a higher or coarser level in the nested hierarchy. Thus, we introduce a region mean-based Potts energy which uses a gradient between region means (as opposed to a gradient between pixels in the classic Potts model)

in order to compute pixel-to-region and region-to-region distances. We use the first principal component of the covariance matrix of the region pixels (essentially the mean direction of the pixels) to represent this mean when using vector angle as a distance measure. We then allow SNA to first aggregate pixels into a region using the edge-based Potts model and carry out processing at coarser scales using the mean-based Potts model. When using a region mean-based model, we are no longer solving the same problem at each level in the hierarchy which makes this process difficult to analyze. However, practical image segmentation results show the importance of this adaptation.

- We integrate within Markov Random Fields the Mixture of Principal Components (MPC) paradigm [34] where regions are defined by the principal component vector corresponding to the largest eigenvalue of the covariance matrix of the data in each region. The new algorithm adapts the MPC framework and as a consequence vector angle to the Markov Random Fields context. Furthermore, the class or region prototypes are determined probabilistically by sampling from a region prototype distribution.

In the domain of developing pixel distance measures, two problems are specifically of interest: physics-based color image segmentation of real world images and phase unwrapping of interferometric Synthetic Aperture Radar (inSAR) images based on image segmentation. With respect to physics- or reflectance-based color distance measures, several contributions have been made:

- Due to the unpredictable behavior of the vector angle distance measure for pixels with low RGB intensities, three new color distance measures are introduced based on a probabilistic interpretation of color in order to create shading invariant and noise resistant color distance measures in RGB.

- Showing that projecting RGB pixels into a 2-dimensional subspace results in a highlight or specularity invariant color space in which a modified vector angle distance measure can be used additionally to achieve shading invariance thus allowing for reflectance-based image segmentation.

- A highlight invariant transformation is applied to one of the three probabilistic shading invariant distance measures in order to create a new probabilistic distance measure that is both shading and highlight invariant.
- Since the vector angle distance measure shows unpredictable behavior when pixel values have very low intensities, a vector angle accuracy criterion that trusts pixel values with high intensity and distrusts pixel values of low intensity is introduced.

Finally, in the domain of pixel similarity for phase unwrapping problems, a new measure for carrying out the segmentation based on phase and coherence maps is introduced. This measure is an approximation of the model-based probabilistic cost function developed in [17]. Furthermore, the application of a Markov Random Field framework to the interferometric synthetic aperture radar phase unwrapping problem using both coherence and phase information is done for the first time in this thesis.

1.4 Thesis Organization

This thesis is organized into nine chapters beginning with a background discussion and proceeding to the contributions in pixel grouping algorithms and pixel distance measures.

Chapter 2 introduces the reader to Markov Random Fields from model specification to optimization algorithms used in obtaining a solution. Chapter 3 presents an overview of pixel grouping methods including spatial aggregation and feature-based clustering, as well as energy-based methods as a separate category. At the end of the chapter, we discuss the myriad of methods that can be used to accelerate Gibbs sampling. Chapter 4 describes the background for pixel similarity distances including the Euclidean distance, the vector angle and other less often used formulations. The Dichromatic Reflection Model used to demonstrate physics-based invariances is introduced and explored. The next four chapters detail the various thesis contributions. Chapter 5 gives a concise description of the stochastic nested aggregation contribution including the Graduated Models strategy. Chapter 6 illustrates the advances in probabilistic distance metrics for color discrimination and introduces a general framework for distance metric derivation based on first principles of the application being considered. Chapter 7 describes advances in clustering- or prototype-

based MRFs. Chapter 8 presents the phase unwrapping problem, and the derived image processing-based solutions. Chapter 9 concludes the thesis and gives recommendations for future research directions.

Chapter 2

Background: Markov Random Field Modelling

Markov Random Fields (MRFs) are a family of models used for large-scale statistical analysis spanning fields such as physics, statistics and computer vision [48, 88]. They provide a convenient and consistent framework for modelling context-dependent entities such as pixels and correlated features through probabilistic distributions of label interactions between neighboring sites or pixels.

Contextual constraints are necessary when trying to interpret visual information. That is, the spatial and visual contexts of the objects in an image scene are necessary for the understanding of the scene; the context of object features at a lower level of representation allow the recognition of the objects; the context of primitives at an even lower level lets the object features be identified; and finally the context of image pixels at the lowest level of abstraction allows for the extraction of those primitives. To create a reliable and effective image analysis system the use of contextual constraints is unavoidable and, therefore, crucial. In this research, special attention will be paid to the lowest level contextual constraints.

The use of the MRF framework in computer vision is very appealing for several reasons:

- Contextual constraints can be easily modelled;
- Effective trade off between local and global constraints;

- A pixel's region membership through the Markovianity property is dependent only on its neighbors (which approximates the whole image) and not on one or more of the previously examined pixels;
- Texture models to enable the segmentation of complex images can be easily integrated;
- There are no special initialization requirements.

The chapter is organized in the following manner. Section 2.1 formulates the graph partitioning problem. The second section discusses neighborhood systems and cliques. The third section explains the Markov Random Fields framework. Section 2.4 presents the Gibbs distribution. Section 2.5 describes commonly used Markov Random Field models for image processing. Local and global optimization methods are discussed in Sections 2.7 and 2.8 respectively. Next, Section 2.9 explains implementation issues. The final section concludes and summarizes this chapter.

2.1 Graph Partitioning Formulation

In order to generalize the discussion to any discrete estimation process on a random field, a graph theoretic framework will be adopted throughout this thesis. Borrowing notation from both [4] and [88], consider a planar adjacency graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, where $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ is the set of nodes that need to be partitioned (*e.g.*, pixels, edges, image features, homogenous image regions, etc.), $\mathcal{E} = \{(v_i, v_j)\}$ is the set of edges connecting adjacent nodes and N the number of all nodes to be partitioned. Alternatively, consider that the graph $\mathcal{G}$ can also be defined on a lattice $\mathcal{S} = \{i \mid 1 \leq i \leq N\}$ which can be a regular or irregular grid. In the case of an image, of width w and height h, the size of the lattice will be $N = w \cdot h$. The image X is composed of pixel values $\{\underline{x}_i\}$ on the lattice $\mathcal{S}$ where pixels are nodes. In essence, $\mathcal{V}$ and $\mathcal{S}$ are equivalent. Both formulations will be used throughout the thesis depending on the emphasis on graphs or images. Note that graph partitioning is also known as discrete state estimation and graph coloring.

The segmentation problem can than be formulated in mathematical terms. If n is the number of graph partitions or regions such that $1 \leq n \leq N$, then an n-partition of the

graph π_n is denoted by

$$\begin{aligned} \pi_n &= (V_1, V_2, \ldots, V_n) \\ \cup_{i=1}^{n} V_i &= \mathcal{V}, \\ V_i \cap V_j &= \emptyset, \forall i \neq j. \end{aligned} \tag{2.1}$$

Initially, each node belongs within one subset such that $V_i = \{v_i\}$. However, through the process of graph partition each subset V_j, $j = 1, \ldots, n$, will be ultimately composed of one or more nodes v_i, $i = 1, \ldots, N$. The process of finding which v_i belongs in which V_j is called graph partitioning in general and image segmentation in the case of images.

In labelling problems such as image segmentation, we assign a label $k \in \mathcal{L}$ to each of the initial subsets V_j for $j = 1, 2, \ldots, n$ where $n = N$. In the discrete case, which is the primary focus of this thesis, a label assumes a discrete value in the set $\mathcal{L} = \{1, \cdots, K\}$. The number of labels is usually fixed, but occasionally it is variable [4]. In this thesis, we consider without loss of generality that each graph vertex $v_i \in \mathcal{V}$ is a pixel or region with a corresponding label[1] l_i which represents a property or feature of the underlying image region or pixel. These features are usually based on intensity, color, texture and other characteristics. Note that one could generalize the concept of label to include model estimation [4, 140]; however, this topic is beyond the scope of this thesis.

The pixels of image X are represented by $\underline{x}_i$ where $i = 1, \ldots, N$. If the graph vertex is a region, then the pixels forming this region are deemed homogenous with respect to some attributes like color or intensity depending on the actual problem definition. The pixel is by definition homogenous as it is the smallest possible component of an image.

Given the partition representation of an image by

$$W = (n, \pi_n, \mathcal{L}) \tag{2.2}$$

where $1 \leq n \leq N$ we would like to obtain the solution to W. We set up an optimization problem maximizing the Bayesian posterior probability $p(X|W)p(W)$ or equivalently minimizing an energy $U(W|X)$ in a solution space Ω,

$$W^* = \arg \max_{W \in \Omega} p(X|W)p(W) \tag{2.3}$$

[1] A label is used to distinguish different nodes from each other locally and is not intended as a global prototype. $l_i = k$ such that $k \in \mathcal{L}$.

or

$$W^* = \arg\min_{W\in\Omega} U(W|X). \tag{2.4}$$

For W in (2.2), Ω_{π_n}, such that $\pi_n \in \Omega_{\pi_n}$, represents the space of all possible n-partitions π_n of $\mathcal{V}$, which leads to the following solution space for W,

$$\Omega_\pi = \cup_{n=1}^{N} \Omega_{\pi_n}. \tag{2.5}$$

In order to search Ω_π (or what we earlier called Ω), we need to determine which nodes should be together and which should not. For discrete labels, this is a finite, but usually very large space of size K^N (in the case of continuous labels, the space would be infinite).

The following sections will give a brief overview of the theoretical foundations of Markov Random Field theory: neighborhood systems, Gibbs distributions, Markov-Gibbs equivalence, MRF models, as well as local and global optimization methods for obtaining a solution. For a detailed account of MRF theory with respect to applications in computer vision please consult [22, 88].

2.2 Neighborhood Systems and Cliques

Before defining a Markov Random Field, it is important to define the notion of the neighborhood system. The neighborhood system is used to relate the sites $\{i\}$ in $\mathcal{S}$ to each other [88]. $\mathcal{S}$ will be assumed to be encoded on a regular lattice or grid (i.e., pixels in an image) as shown in Figure 2.1. This mathematical formulation is necessary in order to encode the desired contextual constraints explicitly. A neighborhood system $\mathcal{N}$ for $\mathcal{S}$ is defined as

$$\mathcal{N} \;=\; \{\mathcal{N}_i \mid \forall i \in \mathcal{S}\} \tag{2.6}$$

where $\mathcal{N}_i$ is the set of sites neighboring i. There are two properties associated with neighborhoods. First, a site i is not a neighbor to itself; *i.e.*, $i \notin \mathcal{N}_i$. Second, the relationship between two neighboring sites i and i' is mutual; *i.e.*, $i' \in \mathcal{N}_i \iff i \in \mathcal{N}_{i'}$.

On a regular grid, the neighboring sites or nodes are usually defined to be those within a radius of r from i such that

$$\mathcal{N}_i \;=\; \{i' \in \mathcal{S} | \Phi_E(i,i') \leq r, i' \neq i\} \tag{2.7}$$

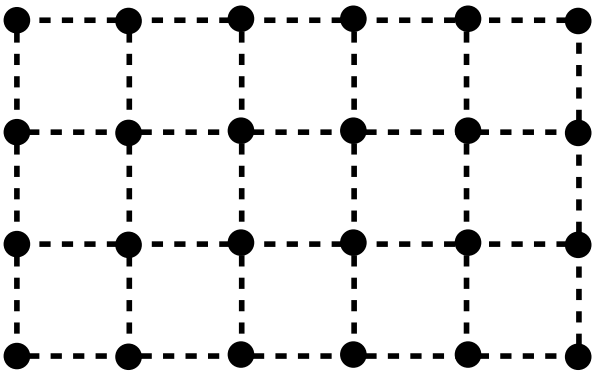


Figure 2.1: A regular lattice or grid.

where $\Phi_E(i, i')$ is the Euclidean distance (1.2) on the grid between the locations of pixels i and i', and not the feature distance as discussed in Section 1.1. Other neighborhoods are possible [31].

For example, Figure 2.2 shows first and second order neighborhood systems which correspond to four-site and eight-site pixel neighborhoods. A zero-site neighborhood would contain the examined pixel and no neighbors. Note that in neighborhoods of order three or greater, the "neighborhood" sites are not all adjacent to the central site.

In addition, the sites at the lattice boundaries have fewer neighbors. For example, for a first order neighborhood system (or four-neighborhood) where $r = 1$ on a rectangular lattice $\mathcal{S}$, the four sites at the corners (bottom-right, top-right, top-left, bottom-left), will have only two neighbors as opposed to the four neighbors for each of the "interior" sites.

Graph $\mathcal{G}$ can also be defined by the pair $\langle \mathcal{S}, \mathcal{N} \rangle$ since $\mathcal{S}$ contains the nodes and $\mathcal{N}$ delineates via the neighborhood relationship the edges between the nodes. To encode a neighborhood structure, we need to define a way to create relationships within that neighborhood. These relationships called cliques for $\langle \mathcal{S}, \mathcal{N} \rangle$ can be defined as a subset of sites in $\mathcal{S}$. For a first order model, they contain either a single-site $c = \{i\}$, or a pair of neighboring sites $c = \{i, i'\}$ [157]. The set of single-site cliques is given by

$$\mathcal{C}_1 \quad = \quad \{i | i \in \mathcal{S}\} \tag{2.8}$$

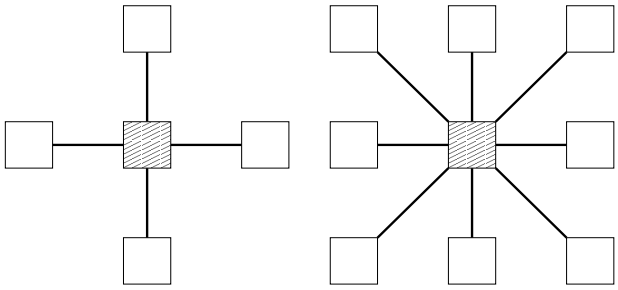


Figure 2.2: Examples of four-site (left) and eight-site (right) neighborhoods usually used in image processing. The black square indicates the central pixel being processed while the white squares are the neighbors.

and correspondingly the set of pair-site cliques is defined as

$$\mathcal{C}_2 \quad = \quad \{\{i, i'\} | i \in \mathcal{S}, i' \in \mathcal{N}_i\} \tag{2.9}$$

Possible cliques for first and second order models are illustrated in Figure 2.3. For a zeroth order MRF, only the clique in (a) would be used. For a first order model, cliques (a)-(c) would be used (only single sites and site pairs). For a second order model, cliques (a)-(e) would be applicable. Compare cliques with neighborhood structures shown in Figure 2.2. As the order of the model is increased, the number and size of cliques rises and processing the model becomes more computationally expensive. In this thesis, a pairwise distance similarity measure will be used which restricts us to first order models with pair-site clique interactions.

In this thesis, we will focus on first order models when defining image segmentation energy models for three reasons: (a) to allow for pairwise contextual constraints (*i.e.*, no constraints for the zeroth-site configuration or for triple-site and higher order cliques), (b) simplification of the joint probability formulation, and (c) lowering computational cost (with respect to the more computationally expensive second or higher order neighborhoods).

Pairwise constraints are very useful for several reasons. First, they limit the computational complexity of the models. Second, it is relatively easy to apply ergodic [48, 4] (one that converges to the stationary probability regardless of initial conditions) optimization algorithms with reversible moves (that can create a path between any two points in a

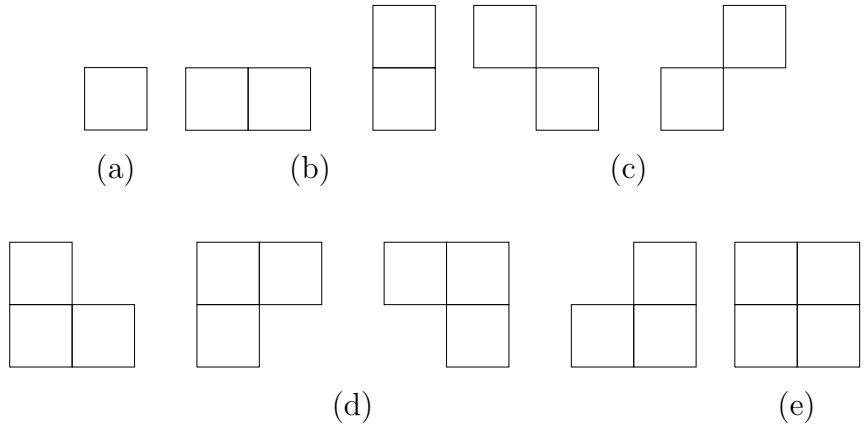


Figure 2.3: Cliques on a lattice of regular sites: (a) single site, (b) horizontal, and vertical pair-site, (c) diagonal pair-site, (d) triple-site, and (e) quadruple-site.

function) to pairwise models. Third, it is easy to analyze them as all interactions are local while in non-pairwise models the interactions could be highly non-local where cause and effect are not easily examined.

2.3 Markov Random Fields

A Markov Random Field can be defined as a family of random variables [48, 88] $\ell = \{l_i\}$ with respect to a neighborhood structure $\mathcal{N}$ on the set $\mathcal{S}$ if and only if the following two conditions are met. First, given that the random variables ℓ will take on a set of values from label set $\mathcal{L}$, the joint probability $P(\ell)$ needs to satisfy $P(\ell) \geq 0$, for all possible label combinations. Second, the joint probability $P(\ell)$ satisfies the Markovianity property [88]

$$P(l_i \mid \ell_i) = P(l_i \mid \ell_{\mathcal{N}_i}) \tag{2.10}$$

where $\ell_i = \{l_j \mid j \in \mathcal{S}, j \neq i\}$ (*i.e.*, all labels except for l_i) and $\ell_{\mathcal{N}_i} = \{l_j \mid j \in \mathcal{N}_i\}$. The Markovianity property allows the formulation of a global optimization problem only in terms of its local interactions. (2.10) provides a framework for conditionally decorrelating the assignment of a label l_i from all other assigned labels in the image and condition this assignment only on the local neighboring pixel labels. Figure 2.4 illustrates this principle.

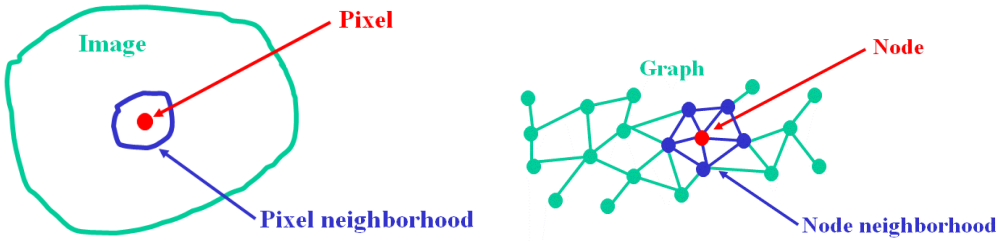


Figure 2.4: (2.10) provides a framework for conditionally decorrelating the assignment of a label l_i from all other assigned labels in the image (left) or graph (right), and condition this assignment only on the local neighboring pixel or node labels respectively.

When applying methods based on Markov Random Fields, two issues become important: defining the joint probability of an MRF $P(\ell)$ and designing an algorithm to find its maximum point or alternatively minimizing the related energy function $U(\ell)$. The optimization algorithms are usually stochastic since the energy function being optimized is typically non-convex (*i.e.*, has many local minima in addition to at least one global minimum).

2.4 Gibbs Distribution

Specifying the joint probability of an MRF, $P(\ell)$, is a difficult if not impossible task for most applications [88]. For a discrete labelling problem, consider that one of K labels is to be assigned to each pixel in an image of size N. Then, the number of possible combinations of labels would be K^N. For typical images of size 256×256 with two labels being assigned the number of solutions is approximately 10^{19728}. If the problem was continuous and therefore the possible labels were the set (or subset) of all real numbers $\mathbb{R}$, there would be an infinite number of possible solutions. However, given a goodness criterion or model for the problem at hand, there would usually be only a few solutions which would be acceptable.

The practical use of MRF models is largely possible due the improved insights and understanding provided by the Hammersley-Clifford theorem [87, 88], which allows Markov random fields to be reinterpreted as Gibbs Random Fields (GRFs) [48, 157]. This theorem

permits MRF problems to be formulated in the context of energy function minimization. Gibbs Random Fields provide a natural way of formulating energy functions to model context dependencies between, for example, image pixels of correlated local features [87]. The second motivating development is the improved insight and available methods for Gibbs sampling [48] (explained in Section 2.6) which can be effectively used to solve GRF/MRF problems.

A Gibbs Random Field is defined as a set of random variables ℓ on $\mathcal{S}$ with respect to a neighborhood $\mathcal{N}$ if and only if ℓ obeys a Gibbs distribution. The Gibbs distribution is defined as

$$P(\ell) = Z^{-1}e^{-U(\ell)} \tag{2.11}$$

where

$$Z = \sum_{\ell} e^{-U(\ell)} \tag{2.12}$$

is the normalizing constant or the partition function assuming discrete random variables (for continuous random variables an integration would be needed). The energy function or model $U(\ell)$ is defined as

$$U(\ell) = \sum_{c \in \mathcal{C}} V_c(\ell) \tag{2.13}$$

where $V_c(\ell)$ are clique potentials over all possible cliques $\mathcal{C}$ for a given model order. In the case of a first order neighborhood, $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2\}$ where $\mathcal{C}_2$ corresponds only to the horizontal and vertical cliques shown in Figure 2.3(b).

Now that we have defined the general structure of an energy model, specific implementations will be considered in the next section.

2.5 Common MRF Models

Several MRF models proposed in the literature are useful for the segmentation of image regions and defining models for textures. The following models will be discussed in this

section [88]: auto-logistic or Ising model, the Gaussian Markov Random Field or auto-normal model [21], and the multi-level logistic model [48] (otherwise known as the Potts or generalized Ising model).

Contextual constraints on two labels are the lowest order constraints to convey spatial information. For a first order MRF, these consist of the cliques in Figure 2.3(b). The constraints are encoded as pair-site clique potentials in the Gibbs energy term, which takes the form

$$U(\ell) = \sum_{i \in \mathcal{S}} V_1(l_i) + \sum_{i \in \mathcal{S}} \sum_{i' \in \mathcal{N}_i} V_2(l_i, l_{i'}) \tag{2.14}$$

or equivalently

$$U(\ell) = \sum_{\{i\} \in \mathcal{C}_1} V_1(l_i) + \sum_{\{i,i'\} \in \mathcal{C}_2} V_2(l_i, l_{i'}) \tag{2.15}$$

where V_1 represents a single-site clique potential and V_2 a pairwise clique potential, while $\mathcal{C}_1$ and $\mathcal{C}_2$ are their corresponding clique sets. Clique functions V_c can take on various forms. An auto-logistic model is considered when $\{l_{i,j}\}$ is represented by a two-value label set such as $\mathcal{L} = \{-1, +1\}$. Then, the energy function used is

$$U(\ell) = \sum_{\{i\} \in \mathcal{C}_1} \alpha_i l_i + \sum_{\{i,i'\} \in \mathcal{C}_2} \beta_{i,i'} l_i l_{i'} \tag{2.16}$$

where α_i is a constant and $\beta_{i,i'}$ represents the pair-site clique interaction coefficients. This is also commonly called the Ising model. Briefly, α_i and $\beta_{i,i'}$ control the relative constraints on the homogeneity and fragmentation due to the contextual constraints and, therefore, how much labels of adjacent nodes or pixels will want to be similar or different from each other. These values *define* the model. Several techniques for their estimation are detailed in [88]. Parameter estimation is beyond the scope of this thesis and, therefore, the models will be determined experimentally. A comprehensive theoretical analysis of these parameters is provided in [74].

An auto-normal model, also called a Gaussian MRF [21], is created when the label set $\mathcal{L}$ is the real line $\mathbb{R}$ and the joint distribution is multivariate Gaussian. These models have been mostly used to describe and synthesize textures [22, 88].

The multi-level logistic model also known as the Potts model is a generalization of the Ising model [48] where there are K discrete labels in the label set $\mathcal{L} = \{1, \cdots, K\}$. The Potts model is formally defined as

$$U(\ell) = \sum_{\{i\} \in \mathcal{C}_1} \alpha_i l_i + \sum_{\{i,i'\} \in \mathcal{C}_2} \beta_{i,i'} V(l_i, l_{i'}) \tag{2.17}$$

where $V(l_i, l_{i'})$ represents the Kroenecker δ. The Potts model will be used extensively in this thesis.

2.6 Gibbs Sampling

Since the solution space Ω is very large, an efficient method needs to be used to explore it. In general, the problem of finding $p(W|X)$ is NP-hard [70] since the relationships between nodes in the graph are not independent. Therefore, several simplifications need to be applied in order to solve it. The main approximation that we will discuss in this section is assuming the Markovianity property is valid for images thereby allowing graph partitioning algorithms for image segmentation to use the MCMC framework.

One common algorithm which performs a Markov chain search and is designed to have a unique invariant (stationary) probability $p(W|X)$ is the Gibbs sampler [48, 88]. The Gibbs sampler generates the next label configuration according to a conditional probability; namely, a candidate $l_i^{(t+1)}$ is drawn randomly from the conditional distribution $P(l_i^{(t+1)} \mid \ell_i^{(t)})$ to replace the existing label $l_i^{(t)}$. Therefore, the transition from $\ell^{(t)} = \{l_1^{(t)}, ..., l_N^{(t)}\}$ to $\ell^{(t+1)} = \{l_1^{(t+1)}, ..., l_N^{(t+1)}\}$ is performed by successively drawing samples form the conditional probabilities which is known as the raster approach [48, 88].

The transitional probability from an initial label configuration $\ell^{(t)}$ to $\ell^{(t+1)}$ is given by

$$P(\ell^{(t+1)} \mid \ell^{(t)}) = \prod_{i=1}^{N} P(l_i^{(t+1)} | l_j^{(t+1)}, l_{i'}^{(t)}, j < i < i') \tag{2.18}$$

where $P(l_i^{(t+1)} | l_j^{(t+1)}, l_{i'}^{(t)}, j < i < i')$ defines the conditional distribution from which each individual node label is drawn. Therefore, the Markovianity property, $P(l_i^{(t+1)} \mid \ell_i^{(t)}) =$

$P(l_i^{(t+1)} \mid \ell_{\mathcal{N}_i}^{(t)})$, can be reformulated as

$$P(l_i^{(t+1)} \mid \ell_{\mathcal{N}_i}^{(t)}) \quad \propto \quad \frac{e^{-\left[V_1(l_i^{(t)}) + \sum_{i' \in \mathcal{N}_i} V_2(l_i^{(t)}, l_{i'}^{(t)})\right]/T}}{Z} \tag{2.19}$$

where T is the temperature parameter which controls the "peakedness" of the Gibbs distribution. Its usefulness will become apparent when discussing stochastic minimization methods in the next section. Through this conditional sampling, the Gibbs Sampler produces a Markov chain $\{\ell^{(0)}, \ell^{(1)}, \ell^{(2)}, \dots\}$ with an equilibrium point corresponding to the joint probability $P(\ell)$ [48, 88]. The Gibbs sampler [48] is given in Algorithm 1.

Algorithm 1 The Gibbs Sampler

1: randomly initialize ℓ to a point in $\mathcal{L}^{\mathcal{S}}$
2: **repeat**
3: **for** $i \in \mathcal{S}$ **do**
4: compute $P(l_i^{(t+1)} = k \mid \ell_{\mathcal{N}_i}^{(t)})$ for all $k \in \mathcal{L}$
5: set $l_i^{(t+1)}$ to k with probability $P(l_i^{(t+1)} = k \mid \ell_{\mathcal{N}_i}^{(t)})$
6: **end for**
7: **until** $\max P(\ell)$

Therefore, label configuration ℓ is chosen according to a stochastic optimization scheme where the selected conditional distribution is not always the most likely (*i.e.*, the chosen configuration has sometimes higher energy). This is in contrast to deterministic optimization schemes where the most likely conditional distribution is always chosen.

Gibbs Sampling simulates ergodic and reversible Markov chain jumps in the space of all graph partitions Ω_π when using a stochastic optimization method like simulated annealing. In other words, it is possible to go from any configuration of labels $\ell^{(t)}$ to any other configuration $\ell^{(t+1)}$ (*i.e.*, no state is a "sink" from which it is impossible to get to another state). Finally, Gibbs sampling is applicable to arbitrary posterior probabilities or energy functions defined on graphs [48, 88].

Other stochastic optimization schemes such as the Metropolis-Hastings [88, 4] algorithm are also possible. The Metropolis-Hastings algorithm works by choosing a label at random and testing whether to accept or reject that label. Its major disadvantage resides in

potentially having to reject many alternatives which can result in getting stuck in an undesirable local minimum. On the other hand, the Gibbs sampler eliminates this problem by searching through all possible labels in $\mathcal{L}$.

Local energy minimization methods will be examined next while global energy optimization methods will be examined in Section 2.8.

2.7 Local Minimization Methods

Besag [7] proposed to maximize local conditional probabilities sequentially due to the difficulty of maximizing the joint probability of an MRF. *Iterated Conditional Modes* (ICM) is a "greedy" algorithm which carries out iterative local energy minimization. ICM works by sequentially updating each $l_i^{(t)}$ into $l_i^{(t+1)}$ by maximizing the conditional (posterior) probability $P(l_i^{(t+1)} \mid \{\underline{x}_i\}, \ell_i^{(t)})$ with respect to $l_i^{(t)}$ given the data or pixels $\{\underline{x}_i\}$ and all other labels $\ell_i^{(t)}$ in the image. Markovianity is assumed as before; namely, that ℓ depends on the labels in the local neighborhood. It follows that we have $P(l_i^{(t+1)} \mid \ell_i^{(t)}) = P(l_i^{(t+1)} \mid \ell_{\mathcal{N}_i}^{(t)})$.

Maximizing $P(l_i^{(t+1)} \mid \ell_{\mathcal{N}_i}^{(t)})$ is equivalent to minimizing the corresponding posterior energy $U(l_i^{(t+1)} \mid \ell_{\mathcal{N}_i}^{(t)})$. For discrete $\mathcal{L}$, $U(l_i^{(t+1)} \mid \ell_{\mathcal{N}_i}^{(t)})$ is evaluated with each possible label $k \in \mathcal{L}$. The label giving the lowest energy value is used to update $l_i^{(t+1)}$. The preceding defines an updating cycle of the ICM when applied to each site i. The iteration continues until convergence to the maximum conditional probability. The convergence is guaranteed for serial updating and is rapid [7].

As widely reported [22, 88], the result obtained by ICM depends very much on the initial label assignment $\ell^{(0)}$. Unfortunately, a proper initialization to obtain a good solution is not known. For example, when trying to restore a noisy image, a natural choice for $\ell^{(0)}$ is the maximum likelihood estimate which corresponds to the actual data when the noise is an identically and independently distributed Gaussian [88]. For image segmentation, $\ell^{(0)}$ can also represent a result obtained by another algorithm such as those in [44].

The problem with local optimization methods (and all deterministic ones in general) is that the label of a pixel (or atomic region) under consideration will be flipped only if a lower energy state can be reached. If a flip is required that will increase the energy in order to bring lower energy in a subsequent step, this will not happen as these algorithms

do not allow this. The only way this could be possible is if the energy model was made to be convex. Essentially, the problem is made convex by moving the non-convexity into a separate pre-processing step. In image processing, this could be done for example by disambiguating weak edges by introducing edge linking [4]. However, these types of operations are certainly error prone as they need a mechanism to decide which edges should be linked and which should not (*e.g.*, one needs to distinguish between edges within a texture which are not very useful vs. edges between distinct regions which are crucial).

2.8 Global Minimization Methods

Global optimization includes stochastic and deterministic methods. For global optimization algorithms the goal is to find the lowest minimum or highest maximum. In general, global methods were devised to deal with non-convex energy functions such as the Potts model since non-convex functions contain many local minima. For convex functions, any local minimum is a global minimum; therefore, global methods are not needed. In this section, we describe several global methods for non-convex functions.

Kirkpatrick's and Cerny's Simulated annealing (SA) algorithm [76, 88] is a stochastic optimization algorithm used for minimizing non-convex energy functions such as those to be used in this research. This method tries to simulate the physical process of annealing when a metal is heated and then slowly cooled down in order to obtain a stronger material (or in computational terms, to find a low energy configuration). Consider a system in which any configuration ℓ (cf. Section 2.4) in the configuration space $\mathcal{L}^N$ has probability

$$P_T(\ell) \quad \propto \quad [P(\ell)]^{1/T} \tag{2.20}$$

where $T > 0$ is the temperature parameter governed by an annealing schedule. T controls the degree of *peaking* in the probability distribution function $P_T(\ell)$. When T is large, $P_T(\ell)$ approaches a uniform distribution on ℓ. For $T = 1$, $P_T(\ell) = P(\ell)$. Choosing T *small* exaggerates the mode(s), thus forcing $P_T(\ell)$ to concentrate on the peaks of $P(\ell)$. As $T \to 0$ according to a schedule of decreasing T values, the distribution takes shape and samples of $P_T(\ell)$ concentrate on the peaks of $P(\ell)$. The SA algorithm is given in Algorithm 2.

Algorithm 2 Simulated Annealing

1: Set T and initialize $\ell^{(0)}$ to a random point in $\mathcal{L}^N$;
2: **repeat**
3: **for** $i \in \mathcal{S}$ **do**
4: Sample l_i from $\mathcal{N}_i$ under T;
5: **end for**
6: **until** $T = 0$
7: Return $\ell^{(t)}$, the last label configuration.

At a fixed T, the sampling for each individual site i is done using

$$P_T(l_i) = \frac{e^{-U(l_i)/T}}{\sum_{k\in\mathcal{L}} e^{-U(k)/T}} \tag{2.21}$$

Usually after the sampling converges to the equilibrium of the joint probability distribution at the current T, T is decreased according to a carefully chosen schedule until it reaches zero. This temperature schedule is usually a major design issue when using simulated annealing. In order to guarantee the convergence to a global minimum (as opposed to a local one) regardless of the initial configuration $\ell^{(0)}$, two conditions are sufficient [48, 88]:

$$\lim_{t\to\infty} T^{(t)} = 0 \tag{2.22}$$

and

$$T^{(t)} \geq \frac{N \times \Delta}{\ln(1+t)} \tag{2.23}$$

where $\Delta = \max_l E(\ell) - \min_l E(\ell)$. It can be shown that, for any given finite problem, the probability that the simulated annealing algorithm terminates with the globally minimum solution approaches 1 as the annealing schedule (2.23) is used [48]. In practice, this implies that the annealing time required to ensure a significant probability of success will usually exceed the time necessary for a complete search of the solution space.

Therefore, usually other schedules are adopted which are a trade-off between performance (how close the algorithm gets to the global optimum) and its convergence speed (how fast it can get there). For example, in [76] the authors choose

$$T^{(t)} = \kappa T^{(t-1)} \tag{2.24}$$

where $\kappa \in [0.8, 1)$ is the typical range of values for the exponential decay. The initial temperature is set high enough in order to essentially accept all configuration changes (*i.e.*, the probability distribution behaves as if it was a uniform distribution). When there are no more changes to the label field, the probabilities are then frozen and annealing stops. In this case, the MRF has converged to the stationary probability $P(\ell)$.

If a suboptimal schedule such as (2.24) is chosen, then it is not possible to expect the algorithm will find the global minimum since one of the sufficient conditions has been broken. However, in practice, optimization results produced using (2.24) are satisfactory. At $T = 0$, the algorithm no longer performs a stochastic search; it becomes a form of deterministic gradient descent. This is a special case of SA called Iterated Conditional Modes (ICM) which is guaranteed to converge to a local minimum [88]. Therefore, simulated annealing can be considered a generalization of ICM.

An important global deterministic optimization method is the *Highest Confidence First* (HCF) algorithm [25]. It is also a serial, deterministic algorithm for combinatorial minimization for discrete label sets. Its main feature is assigning labels first to regions which when labelled would decrease the overall energy the most, hence the name of the method. HCF introduces a special *uncommitted* label and the strategy for *committing* a site. The uncommitted label is denoted by 0. Thus, $\mathcal{L}$ is augmented into $\mathcal{L}^+ = \{0, \mathcal{L}\}$. A label l_i is said to be uncommitted if $l_i = 0$ or committed if $l_i \in \mathcal{L}$. Initially, all site labels are defined as uncommitted, *i.e.*, $\ell^{(0)} = \{0, 0, \ldots, 0\}$. Once a site has been committed, its label cannot be changed back to 0; however, it can be updated to another value in $\mathcal{L}$.

The same energy as in ICM, $U(l_i^{(t+1)} \mid \ell_{\mathcal{N}_i}^{(t)})$, is minimized. However, in order to produce an ordering of which sites should be examined first, the stability of i with respect to l_i is computed for all nodes i such that:

$$S_i(k) = \begin{cases} -\min_{k \in \mathcal{L}, k \neq l_{min}} [U_i(k) - U_i(l_{min})] & \text{if} l_i = 0 \\ \min_{k \in \mathcal{L}, k \neq l_{min}} [U_i(k) - U_i(l_{min})] & \text{otherwise} \end{cases} \tag{2.25}$$

where $l_{min} = \arg\min_{k \in \mathcal{L}} U_i(k)$. The stability S_i of an uncommitted site is given as the negative difference between the lowest and the second lowest conditional energies. The stability S_i of a committed site is the difference between the current local energy $U_i(l_i)$ and the lowest possible energy due to any other label. The stability range is $-\infty < S_i < +\infty$. A negative stability indicates that the energy could be possibly lowered. All uncommitted

sites have non-positive S_i. The magnitude of S_i is equivalent to the change in energy given by the change in the label l_i. A lower value of S_i indicates a more stable configuration while a negative value with a larger magnitude increases the confidence to transform $\ell^{(t)}$ into a new configuration $\ell^{(t+1)}$.

The order of the update depends on which site is least stable, *i.e.*, has the lowest S_i. At each step, only that particular site is allowed to change its label. Suppose that $s = \arg\max_i S_i(l_i)$ is the least stable site. Then if $l_s = 0$, change l_s to $l'_s = \arg\min_{k \in \mathcal{L}} U_s(k)$ otherwise change l_s to $l'_s = \arg\min_{k \in \mathcal{L}, k \neq l_s} [U_s(k) - U_s(l_s)]$. Therefore, the first committed label l_i is the one which corresponds to the maximum local likelihood.

For each iteration, the number of different merging configurations tested is approximately $O(N)$ since some results from the previous iteration can be reused in the next. At each iteration after the first, the algorithm needs to compute only the pairwise merging cost between all groups and the newly-merged group from the previous iteration resulting in an overall complexity of $O(N^2)$ [121].

HCF is an enhanced version of ICM that might prevent the optimization getting stuck in some local minima. It also appears to be faster than either ICM or SA given that it only requires (in practice) on average just over one iteration through the data to converge to a solution for a small graph [94] which might not be the case for very large graphs. However, HCF is still a deterministic method which might not be able to reach a local minimum that is close enough to the global minimum.

There are also several global deterministic optimization methods such as Graduated Non-Convexity (GNC) [9] and Deterministic Annealing [26, 27, 65]. Graduated non-convexity is a deterministic annealing method which approximates the global solution for non-convex minimization of unconstrained, *continuous* problems. The basic principle behind GNC is the following. Consider that we would like to find the optimum point of a non-convex function. The energy function $U(\ell)$ is made convex through application of a parameter γ which is set to a sufficiently large value to make $U(\ell|\gamma)$ strictly convex. The minimum of this function can be easily found using standard greedy methods (*e.g.*, ICM) regardless of the initial conditions. The minimum found under the first γ, say $\gamma^{(0)}$ is used as the initial value for the next minimization under $\gamma^{(1)}$ and so on. $\gamma^{(t)}$ for each new minimization t is then gradually relaxed until the original non-convex function is reproduced.

If the local minima found for each $\gamma^{(t)}$ are tracked, it may be possible to find a solution close to the global optimum of the original non-convex function.

2.9 Implementation Issues

There are several implementation issues: raster vs. random scanning, quality of random number generators, continuous vs. discrete MRFs, and non-regular lattices.

In MRF algorithms, no two neighboring sites should be updated simultaneously [88] since the Markovianity property assumes that the neighboring labels l_i are conditionally independent. Thus, random scanning is better than raster scanning since in raster scanning only the first visited node would be conditionally independent from the other nodes whereas in random scanning the conditional independence would be violated only when two adjacent nodes are examined one after the other. A more robust method is needed to ensure this conditional independence.

The "coding method" [6] may be incorporated into Gibbs Sampling to parallelize the scan and thus force conditional independence. The coding method partitions $\mathcal{S}$ into several disjoint sets $\mathcal{S}_{(f)}$, called codings, such that no two sites in one $\mathcal{S}_{(f)}$ are neighbors. Figure 2.5 illustrates the codings for the 4-neighborhood system where $\mathcal{S}_{(1)}$ and $\mathcal{S}_{(2)}$ are needed akin to a checkerboard set up with squares of 1's and 2's. Four codings are needed for the 8-neighborhood system. In general, any neighborhood system can be implemented using the coding method. This leads to all l_i on a single coding $\mathcal{S}_{(l_i)}$ to be updated in parallel.

The fact that the sites within each $\mathcal{S}_{(f)}$ are not each other's neighbors provides computational advantages especially in a parallel processor implementation. Under the Markovianity assumption, the variables associated with the sites in an $\mathcal{S}_{(f)}$, conditioned on the labels at all other sites, are mutually independent.

An important issue is the quality of random number generators used to carry out the sampling operation. Random numbers produced by computers are usually generated using pseudo-random number generators. These programs produce long sequences of *quasi* random numbers based on some initial seed [144]. A random number generator based on deterministic computation is not viewed as a *true* random number generator since by

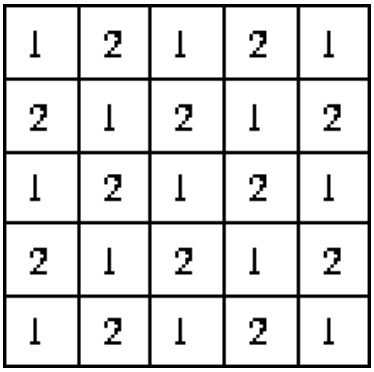


Figure 2.5: Codings for the four-neighborhood system. Pixels marked number k belong to the coding $\mathcal{S}^{(k)}$.

definition its output is predictable. However, simple pseudo-random number generators[2] can be used instead of true random numbers in many applications as long as the period of the generator is less than the total number of produced random numbers. In this thesis, we use a simple pseudo-random generator with a period of 2^{31}.

In this thesis, we will examine discrete MRFs. In some cases continuous MRFs will also be used; however, ultimately, their values will be quantized in order to transform them into discrete MRFs. The main reason for this solution is the much higher complexity of carrying out Gibbs sampling on continuous state MRFs than on discrete state MRFs. Discrete MRFs have usually a large but *finite* solution space which is much easier to examine and thus makes the problem easier to solve. For Gaussian MRFs, where continuous parameters are estimated, the problem is tractable and has been addressed in the texture literature [22].

Finally, all of these methods can be applied to image patches or regions that have been obtained by some preprocessing method. For example, regions could be obtained with respect to color [119] or filled in with a texture feature [21] or delineated with via the Canny edge detector [15] and described by histograms [4]. In those cases, the neighborhood grid is irregular. Figure 2.6 shows an irregular grid.

[2]We are using the `rand() function in C`.

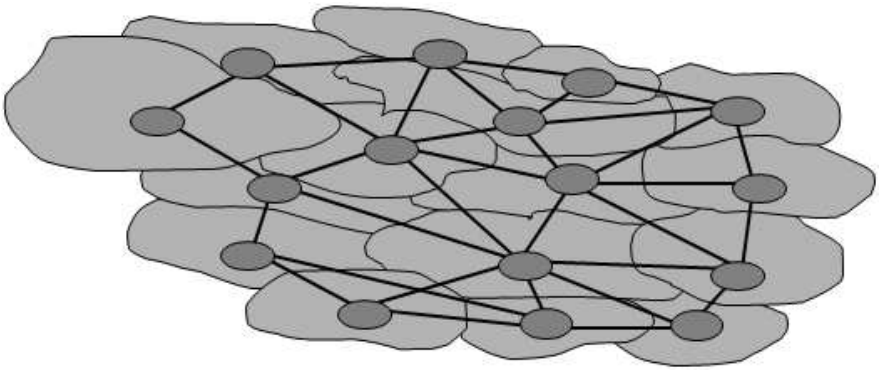


Figure 2.6: An irregular grid such as the one shown here is usually applicable to any underlying structure that is not regular such as pre-segmented image patches or blobs.

2.10 Summary

Markov Random Fields provide a means of conditioning the dependence of the label at one site on its neighbors rather than on the whole label field. This allows graph partitioning problems to become tractable in a stochastic framework. Since we are concerned with image segmentation problems, we will focus on piece-wise constant functions such as the Potts or multi-level logistic models which are characterized by large areas of constant labels (*e.g.*, image regions homogenous with respect to some features). A good local minimum of such functions can be obtained using a stochastic optimization approach such as simulated annealing.

There are two related issues. First, simulated annealing is in theory guaranteed to reach a global minimum; however, in practice it is not possible to run this algorithm with the given parameters (T schedule) as the computational complexity is too great. Second, iterated conditional modes has no computational issues; however, it converges to a local minimum even with very good initial conditions $\ell^{(0)}$. It will be shown in Chapter 5 that merging the idea of irregular neighborhoods with Gibbs Sampling can lead to solving both problems with one crucial assumption which can be satisfied with a constraint on models.

Chapter 3

Background: Review of Pixel Grouping Algorithms

In general, image segmentation can be accomplished by:

- finding regions uniform with respect to some homogeneity criterion (e.g., color, intensity, texture),
- finding the boundaries between different regions,
- using both strategies.

In this thesis, the first paradigm will be examined within the Markov Random Field framework. Boundary or edge detection methods are explored in detail elsewhere [79]. However, edge detection methods, and gradient computation in particular, can be used as distance measures and will be discussed briefly later in this chapter. Extensive work has been done on image segmentation. Readers are encouraged to examine past survey papers on grayscale image segmentation [44, 58, 105] and color image segmentation [5, 24, 93].

The goal of this chapter is to survey the design and implementation of the algorithms most commonly used for image segmentation in general and color image segmentation in particular. Color image segmentation algorithms are considered in detail since most of this thesis will use color images for testing the algorithms presented herein. Algorithms used for graph partitioning which can be easily adapted to an image segmentation task are

also examined. Note that many of these algorithms are not suitable directly for real-world applications. In particular, problem-context sensitive pre- and post-processing steps may be necessary, as well as parameter estimation or tuning. Thus, this chapter seeks to serve as a stepping stone illustrating the algorithms and concepts at the heart of color-based image segmentation systems and not describing full systems.

3.1 Introduction

Haralick and Shapiro argue that there can be no full theory of clustering and, therefore, no full theory of image segmentation [59] since the types of regions to be extracted are predicated on the objectives pursued in each application and, therefore, on the kind of technique used for the segmentation. Subsequently, there exists only a series of different segmentation techniques, each of them with benefits and drawbacks.

This means that image segmentation techniques are generally *ad hoc* and differ on how they emphasize one or more of the desired image properties. Haralick and Shapiro proposed four guidelines for the general image segmentation problem [59]. The criteria can be reformulated in the following way for any type of image:

1. Each segmented regions should be uniform with respect to a set of features,
2. Adjacent regions dissimilar in features should be separated,
3. Larger regions should not contain many small regions within, and
4. The boundaries of each region must be spatially accurate.

In this thesis, the first two criteria are adopted given their general presence in the image segmentation literature as feature discriminators or distance measures. These distance measures generate a small distance between highly similar regions and a large distance between dissimilar ones. We will review color distance measures in Chapter 4. The third criterion indicates that over-segmentation should be minimized as much as possible through region size constraints, for example [39, 142]. Our Bayesian segmentation framework clearly incorporates the first two of Haralick and Shapiro's criteria. The third criterion can be incorporated through additional constraints on the size of each segmented region V_i, an

issue which we will not address in this thesis (see [39, 142]). Since the concept of "spatial accuracy" is ill defined, we will not consider it either.

A huge number of alternative techniques are possible, most problem-context dependent. The structure of the algorithm will determine how the pixel grouping and therefore the image segmentation is carried out. For example, many contexts have specialized initialization requirements, the quality of which can significantly influence results. Next, algorithms may operate in a sequential manner such that the decisions at a pixel may depend on previously visited pixels as opposed to simultaneous operation (or non-causal) algorithms which are independent of the sequence of pixel examination [48, 88, 157]. The distance measure may be point-wise, spatially local, region-based or global. Finally, many algorithms use auxiliary information to improve the segmentation result: edge maps [2, 4, 119, 168], texture models [4, 21], region size [39, 142], minimum or maximum number of regions [35, 69], etc.

Out of these possibilities we consider three general classes of segmentation algorithms, which parallel the organization of this chapter: clustering [35, 69], spatially-based [59, 138], and energy minimization [48, 88, 157, 167]. Clustering algorithms aggregate pixels based on features alone thus they are usually simultaneous with point-wise/global constraints. Section 3.2 summarizes the state of the art in clustering and point-based algorithms. Spatially-based methods, described in Section 3.3, use local pixel relationships to segment images, and are normally sequential in nature with local (and possibly global) constraints. The spatially local nature makes it easy to incorporate edge information. Finally, energy minimization is presented as a separate section as the methods, discussed in Section 3.4, provide a probabilistic framework for simultaneously grouping pixels with local (and sometimes global) constraints, with easy means to incorporate texture and edge information, but with usually considerably increased computational requirements. Finally, Section 3.5 describes various top-down and bottom-up hierarchies that have been used to decrease the computational complexity of many energy minimization-based algorithms.

3.2 Clustering

Of the three fundamental approaches to segmentation, this section explores the use of clustering methods. If a number of objects in an image can be distinguished on the basis

of color, or other feature then the pixels associated with these objects should lie in tight clusters in their associated three-dimensional color space or some feature space respectively. If these clusters can be identified, then the segmentation problem is solved: each image pixel is labelled or grouped by its associated cluster.

Clustering (also known as vector quantization) has been widely explored in the pattern recognition literature, so a wide variety of algorithms is available [35, 69]. The main advantages of clustering are its computational speed, that it aggregates with respect to global features, and that it will separate differently colored regions regardless of their spatial location. The main disadvantages are that spatial information is not taken into account, thereby possibly failing to create spatially compact regions, many small spurious regions may occur in noisy images and the number of clusters must usually be specified *a priori* (the cluster validity problem [35]). Obviously, extensions to standard clustering have been developed to ameliorate these difficulties. Several researchers have devised problem-specific solutions to estimate the optimal number of clusters [115, 150], and rather than clustering just single pixels, the clustering can be applied to pixel neighborhoods (*e.g.*, a 3-by-3 window) effectively embedding a spatial regularizer or smoother within the algorithm [148, 149]. In addition, a class of clustering algorithms exists which also includes pixel position in its feature space [18, 39].

The easiest way to identify a cluster is by defining a prototype (typically its center or mean) such that $\underline{w}_k = \frac{1}{N_k} \sum_{j \in V_k} \underline{x}_j$ where N_k is the size of regions V_k. Non-prototype based approaches [35] such as k-NN (where k represents the number of neighbors) and hierarchical clustering are also possible but will not be discussed in this context due to their computational complexity (especially for $k \gg 1$) and no significant results respectively. The classical prototype-based clustering problem is defined in Algorithm 3. In this case, the pixels in one region can be spatially separated.

Algorithm 3 The clustering algorithm

1: Given pixels $\{i\}$ with their corresponding values $\{\underline{x}_i\}$ and K, the number of both labels and regions,
2: Let V_k be clusters of pixels or nodes each with a prototype $\underline{w}_k$,
3: Then find $\{\underline{w}_k\}$ minimizing some distance criterion.

The most widely used prototype-based algorithm for clustering is k-means [35] where k corresponds equally to the number of labels and regions (each region has an associated label and prototype). It is shown in Algorithm 4. k-means is based on iterative pixel-to-prototype distance computations using the Euclidean distance (1.2) and prototype definition based on the vector mean of pixels in each region or cluster. The main premise of the algorithm is to try to minimize the variance within each cluster. The main advantages of the algorithm are its low computational complexity and ease of analytical analysis (not usually the case for other clustering algorithms). The main disadvantage, however, is the low likelihood of converging to a global minimum since the iterative algorithm uses gradient descent [35] to obtain the solution from random starting points. The solution to each algorithm trial is highly dependent on the starting points. Furthermore, for color image segmentation problems, it is not necessarily desirable to apply this algorithm *as is* since the color space used is not necessarily perceptually uniform; in other words, distances between colors are not reflective of the perceived distance between them when viewed by a human observer [68] (a desirable feature when using the Euclidean distance).

Algorithm 4 The k-means algorithm

1: Given pixels $\{i\}$ with their corresponding values $\{\underline{x}_i\}$ and K the number of labels and regions,
2: Let V_k be clusters of pixels or nodes with a mean$\underline{w}_k$, such that $V_k = \{i \mid \Phi(\underline{x}_i, \underline{w}_k) < \Phi(\underline{x}_i, \underline{w}_l), \forall\ l \neq k\ \}$,
3: Then find $\{\underline{w}_k\}$ minimizing $\sum_{k=1}^{K} \sum_{n \in \mathcal{W}_k} \Phi(\underline{x}_n, \underline{w}_k)$.

Segmentation using k-means has been undertaken in a variety of color spaces: CIE *Lab* [148], using the Dichromatic Reflection Model (DRM) to obtain shading invariance [134], a k-means-like algorithm based on the DRM for highlight and shading invariant segmentation [125], and other k-means variations to cluster pixels in 2-D chromatic and 1-D intensity space [92, 112]. ISODATA is a common generalization of k-means allowing cluster splitting and grouping [35].

In the limiting case where the prototypes $\{\underline{w}_k\}$ are predetermined on a regular grid, the clustering reduces to the special case of histogram analysis. Because the number of histogram bins K grows rapidly with the number of dimensions, histogram analysis is

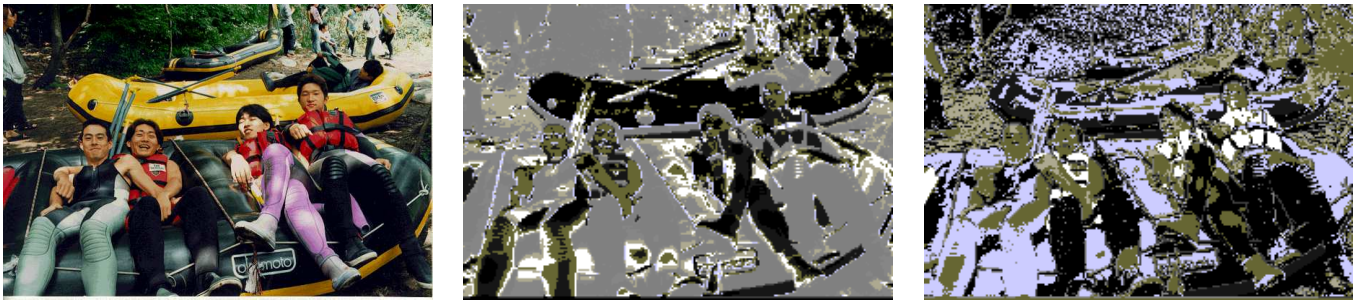
limited to 1-D, 2-D or possibly 3-D spaces, in contrast with general clustering approaches, which handle high dimensional data (which would be the case for multispectral images [125, 136] or texture features) with ease. Histogram-based segmentation methods include the work of Schettini [120] (a recursive one-dimensional histogram analysis), and Kurugöllü and Sankur [81] (analysis of 2-D histograms of RGB pairs RG, GB and RB). More detail on other histogram-based methods can be found in [93].

A very common variation of k-means is the fuzzy c-means algorithm [8]. This algorithm differs from k-means by creating *degrees* of membership in any one cluster as opposed to a pixel being forced to be in a unique cluster. As a consequence calculating the distance between the pixel and prototype needs to be done using a fuzzified version of the Euclidean distance (1.2):

$$\Phi_F(\underline{x}_i, \underline{w}_k) = (u_{ki})^\theta (\underline{x}_i - \underline{w}_k)^T (\underline{x}_i - \underline{w}_k) \tag{3.1}$$

where u_{ki} represents the membership value of the i-th data sample for k-th cluster, and the weighting exponent θ is added for the fuzzification of memberships. The Euclidean distance could be replaced in theory by other distance measures. The larger θ is, the fuzzier the memberships are (usually $\theta > 1$). The sum of u_{ki} with respect to k is constrained to be 1 (for possibilistic approaches this constraint is relaxed [37]). Fuzzy memberships allow pixels to belong to several classes (with different degrees) at the same time with usually one predominant class. This relaxes the rigid cluster boundaries by creating a user defined (based on fuzzy memberships) "gray" zone in the feature space. Lim and Lee [89] apply the fuzzy c-means algorithm to RGB images. Wu *et al.* use a fuzzy c-means clustering algorithm for preliminary segmentation of RGB map images for the purpose of extracting lines and text [159]. A detailed comparison between the k-means and fuzzy c-means approaches is reported in [114].

Clustering using Gaussian mixtures is a probabilistic variant of k-means [35, 131]. In this case, each cluster is one component of the mixture. The pixel distribution in each cluster is typically defined to follow a Gaussian distribution with a mean (the cluster prototype) and a covariance matrix (defining the probabilistic relationship between pixels within that region). The distance measure in this case is the Mahalanobis distance (cf. Section 4.3.2) and the parameters of the mixture components that need to be updated



(a) (b) (c)

Figure 3.1: Different distance measures for clustering can greatly affect the segmentation result: (a) original Image, (b) clustering with k-means (Euclidean distance) using $k = 8$, (c) applying Mixture of Principal Components (vector angle) with $k = 8$. Note that the segmentation of skin pixels with MPC is much more consistent than with k-means.

are the mean and the covariance matrix. This alternative has been used mostly in energy minimization approaches [166] discussed in Section 3.4.

The Mixture of Principal Components (MPC) algorithm [34] is similar in structure to k-means with two fundamental differences. First, distances between the prototype and pixels are computed using the vector angle (1.3) instead of the Euclidean distance (1.2). Second, each prototype is represented by the first principal component (the principal component corresponding to the largest eigenvalue) of the covariance matrix of the pixels belonging to the cluster [148]. In this way, the average direction of the cluster is obtained and not the vector average of the color pixels. This is needed since the vector angle captures only chromaticity-related information which is characterized by the pixel direction in RGB and not its magnitude. A comparison between k-means and MPC in various color spaces is presented in [149, 153]. See Figure 3.1 for an illustration of using two different distance measures in image segmentation.

Clearly all of the above clustering strategies can be generalized from the clustering of 3-element vectors encoding a single color to more general notions of features, which could include information about texture [18], pixel location [18, 111] or surrounding pixel values [148]. Park *et al.* use morphology to expand color clusters nonlinearly in feature space [111].

Several other methods make use of a clustering approach as a preliminary step to more specific region segmentation or merging. For example, Hedley and Yan [64], as well as Zhou *et al.* [165], use distance measures which incorporate a spatial component to complement their color clustering approaches in order to mitigate the effects of outliers and anti-aliasing.

Clustering algorithms are very good at partitioning an image into globally distinct colors. However, they do not always perform well spatially given their ignorance of local relationships. The next section will describe spatially-based algorithms which rely on local pixel information to segment images.

3.3 Spatially-Based Methods

Spatially-based image segmentation algorithms [59, 92, 137] are based on the premise that local information is crucial in the image segmentation process. Whereas in clustering-based approaches there is no guarantee of spatial compactness, the goal of spatially-based methods is to distinguish objects in an image on the basis of pixel adjacency and feature similarity. The pixels associated with an object should be next to each other in the image and have similar characteristics with respect to the chosen distance measure Φ. Therefore, what we seek are *spatially compact regions* that should also be compact clusters in feature space. If these two goals can be achieved, spatial regions can be identified and, therefore, the segmentation problem is solved: each image pixel is labelled or grouped into its associated spatial region homogeneous with respect to some features.

The main advantages of spatial methods are the generation of spatially compact regions and their ability (in most cases) not to depend on the cluster number specification (as is not the case in clustering). The main disadvantages are the ambiguity of selecting starting *seed points* [1, 59] and region-to-region spilling [138].

By definition, a region growing algorithm needs to start the process of growing pixels into regions at a single point, usually called a seed. The algorithm then attaches other pixels to the seed based on some similarity criteria. Seed points may be manually or automatically selected points [1] or even all pixels in the image. When regions are grown by sequentially adding *new* pixels to the region, it is possible that they absorb multiple seed pixels in the growth process. After all seeds have been visited, it is possible that some

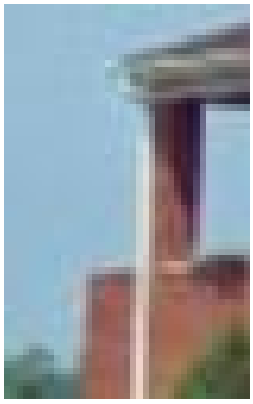
pixels lie outside of all regions, in which case a post-processing step may be necessary to determine whether a given unlabelled pixel should be added to an adjacent region, used as a new region seed, ignored, etc. The biggest disadvantage of these points is that some regions might not be detected if there is no seed within to grow into the full region.

An equally serious issue is region-to-region spilling which may occur when two different regions, connected by a slowly changing gradient, are joined. This occurrence is illustrated in Figure 3.2. In general, a gradient lower than a threshold means that the pixels belong to the same regions while a higher gradient value separates them. If there is one low gradient between two pixels even though all other gradients are high, the pixel will be merged with the surrounding region.

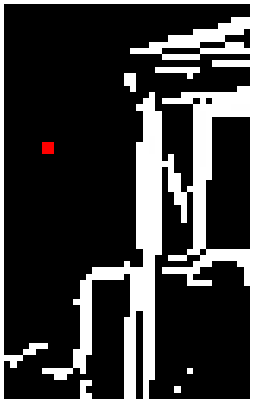
Three classical spatial image segmentation algorithms are reviewed: region growing [137], split-and-merge [59], and edge-based methods such as the watershed [122]. An earlier comprehensive overview of spatially-based techniques may be found in [93]. Region growing algorithms have been very popular due to the simplicity of their implementation and intuitive appeal. Trémeau and Borel [137] use region growing in RGB followed by a merging step. Furthermore, Trémeau and Colantoni propose an adjacency graph to enhance region growing and watershed transform-based segmentation algorithms [138]. Maxwell and Shafer propose a physics-based method using multiple hypotheses of image formation for images without [96] and with highlights [97]. Another type of region growing method creates connected components [146, 147] based on distances between the candidate pixel and an adjacent pixel belonging to the region, and between the candidate pixel and the region prototype that are both less than some experimentally set thresholds. The region prototype is determined by computing the vector mean of the pixels within the region. The similarity measure is the Euclidean distance as in [137] which in [146] applied to a five-plane combination of the XYZ space and the normalized uv planes. Region-to-region spilling [1] is evident in Figure 3.3. A sample region growing method is shown in Algorithm 5.

Split-and-merge methods operate on the dual basis of region splitting and region merging [59]. The splitting phase may be an initial partition of an image [50, 77, 78], or an initial segmentation of the image [59]. The aim is to obtain regions that are homogeneous

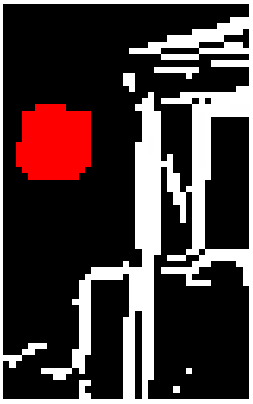
[1] Results taken from [151] with two different parameter sets. Method details in Chapter 7.



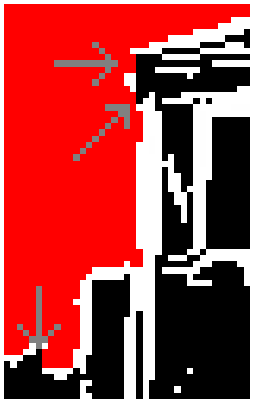
Original Image



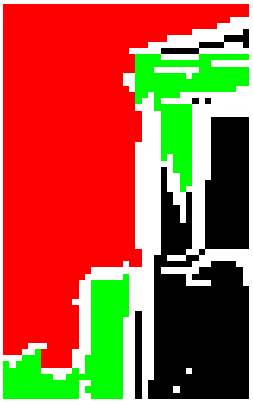
Seed Pixel (red)



Region Growing



Three Critical Points



Region-to-Region Spilling Has Occurred (green)

Figure 3.2: Region-to-region spilling may occur when two different regions, connected by a slowly changing gradient, are joined. Starting with some seed pixel, the region growing algorithm gathers pixels closest to it until it reaches some critical points (*i.e.*, edges between distinct regions with weak gradients) which are then breached. White pixels indicate strong edges between regions (gradient magnitude higher than a threshold) while black pixels show that there is no edge (gradient magnitude less than a threshold). Red pixels illustrate the entire region growing process from starting with a seed pixel to filling in the whole region.

Algorithm 5 A Sample Region Growing Algorithm

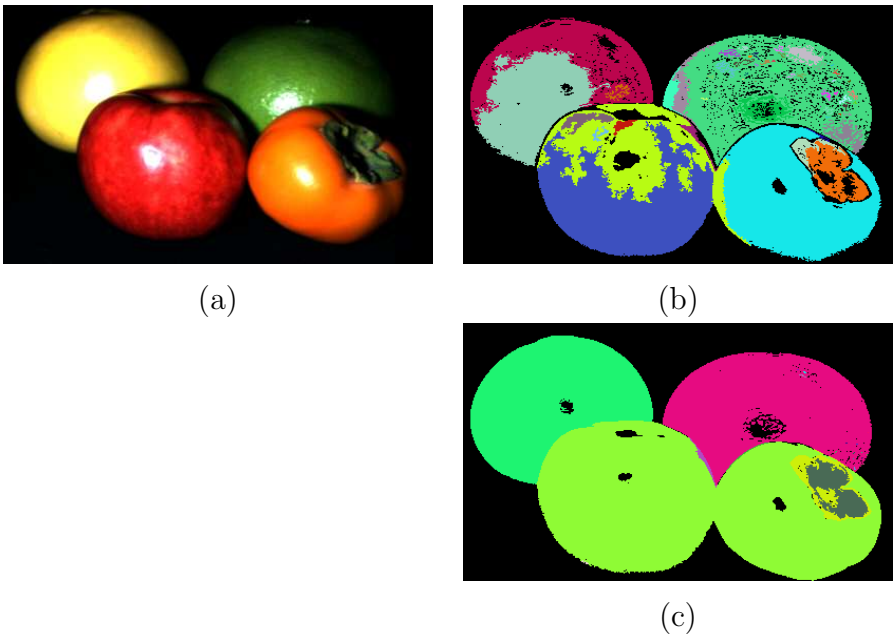
1: Identify a set of initial or seed points $\{s_i\}$ where $i = 1, \ldots, N_s$ and N_s is the total number of seed points such that $N_s \leq N$.

2: **for** All seed pixels $\{s_i\}$ **do**

3: Grow regions $\{V_k\}$, each from s_i, by adding adjacent pixels similar to the pixels already in the region:

$$\underline{w}_k = \begin{cases} s_j \\ \{V_k, j \mid \text{given } i \in V_k \; \exists \; i, j \text{ adjacent}, \forall \; \Phi(\underline{x}_i, \underline{x}_j) < \tau\} \end{cases} \tag{3.2}$$

4: **end for**



(a) (b) (c)

Figure 3.3: Region-to-region spilling result: (a) original image, (b) and (c) region growing results using different sets of parameters. In (b), the thresholds for including a pixel in a region are very conservative leading to oversegmentation with many small regions while in (c) the parameters are much more relaxed leading to undersegmentation with few regions resulting from region-to-region spilling, as happens with the orange and red fruits in the foreground, colored a single green shade.

with respect to the chosen criterion Φ. The merging phase is concerned with joining similar adjacent regions. These algorithms can be implemented as a two step process or an iterative one running until there is convergence to some segmentation result. Klinker *et al.* [77, 78] determine the principal components of image patches to estimate the shading-invariant (with respect to the DRM [123]) region color. Patches of similar color are then merged to form the segmented regions. Gevers [50] implements an incremental Delaunay triangulation scheme for neighborhood referencing instead of a quadtree structure. The image is split using edge information until all triangles satisfy a homogeneity criterion. Next, a merging phase is applied in which the same homogeneity criterion is used to merge adjacent triangles.

The watershed transform works by considering an image a topographic surface based on feature gradients [45, 122, 141]. Then, the minima of this relief can be "pierced" and the image immersed into a "liquid." As the liquid floods the image, if we prevent the merging of the liquids coming from different sources/minima, we partition the image into two different sets: the catchment basins – homogenous image regions – and the watershed lines – the object edges. In other words, the watershed transform could be considered to be region growing coupled with edge detection. Shafarenko *et al.* describe a method for color texture segmentation [122]. They define a color gradient to measure color similarity in CIE *Luv* space. Gao *et al.* use mathematical morphology followed by a modified watershed transform to segment images in CIE *Lab* [45]. Vanhamel *et al.* use a method which controls oversegmentation using a multiscale framework [141].

Edges have been used as an alternative to image segmentation methods since in edge detection the actual boundaries between objects are being sought [59]. Researchers have long taken advantage of many well established methods [54, 59] to provide additional context and guidance for spatial methods. For example, [38] applied an edge detector to an image followed by region growing, where seeds for the region growing algorithm are the centroids of the initial edge map. Ideally region boundaries will coincide with detected edges; if not, the adjacent regions are examined for merging or splitting criteria. Thirion *et al.* [130] proposed another method which combines regions and edge information in a Dichromatic Reflection Model framework to detect pipes. First, to detect highlights, they compared each pixel to the nearest previously calculated linear pipe cluster. Second, the

anisotropy of the image gray levels at each pixel is calculated to obtain shading invariance given that the shading of pipes is highly anisotropic: almost constant along the axis of the pipe and strongly varying in the orthogonal direction. Finally, they used a Canny-Deriche filter to detect edges followed by a contour closing step. In [168], edges were extracted using a gradient operator which is followed by a region growing step called pixel fusion.

Finally, in JSEG [32] two independent steps are applied: color quantization and spatial segmentation. First, the authors use a color image quantizer in CIE *Luv* to obtain representative color clusters in the image. Next, a segmentation criterion is applied to determine boundaries and interiors of color-texture regions using a region growing method. Another method uses color watersheds for spatial information extraction and color region prototypes (obtained with k-means or a Bayesian classifier) for global features [85].

Typically, region growing methods are the simplest to implement and have the least computational complexity while split-and-merge methods are usually iterative. Region growing and watershed methods might be also much more prone to error since seed pixels need to be carefully selected. Watershed and edge-based techniques are also potentially much better than other methods due to their integration of edge information. However, it is clear that the JSEG algorithm presents some clear advantages to other methods due to its inclusion of spatial and color features [32]. Heuristics-based combination of clustering and spatial methods may work in some cases; however, a principled approach to such a combination is preferred. This is the topic of the subsequent section.

3.4 Energy Minimization: Energy Models

One popular framework for describing complicated probabilistic problems is energy minimization, a term stemming from its mathematical origins in statistical physics [48]. This approach to image segmentation can be formulated using elements from both spatially-based and clustering-based approaches enhancing the advantages of both paradigms while limiting the drawbacks some of which are illustrated in Figures 3.4 and 3.5. The relative strengths and weaknesses of the three segmentation algorithm categories are summarized in Table 3.1. The critical strength of energy-based or probabilistic approaches lies in the stochastic optimization formulation whereas the methods described in the previous two

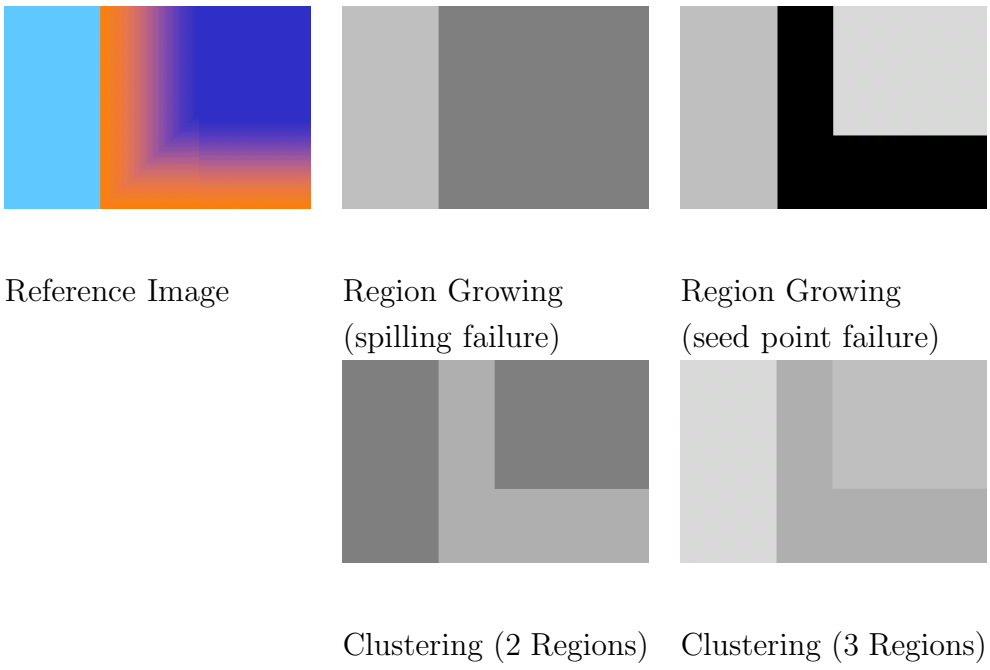


Figure 3.4: An illustration of spatial method failures. In region growing, region-to-region spilling (the dark blue and orange regions are merged) and not choosing good seed points (the "black" colored region has not been detected since it is missing a seed point) are often problematic issues. Using a clustering method, the reference image can be correctly segmented by choosing the appropriate number of regions to segment. However, the correct number of regions is only known after the segmentation has occurred (this is the cluster validity problem). Here both clustering results would be acceptable.

sections were ad-hoc and did not have a well-defined objective function with an optimum point.

Energy minimization problems in computer vision such as image correspondence [4] and image segmentation are long-established [48, 88, 157, 167]. These methods are usually based on a Bayesian framework as defined in Chapter 2 [4, 88]. Computer vision problems are formulated in such a way as to partition image elements, represented by vertices or nodes on an adjacency graph, into larger spatial structures in order to optimize a Bayesian posterior probability or energy function usually derived from statistical physics.

There are two primary concerns: how to define an objective function for the image

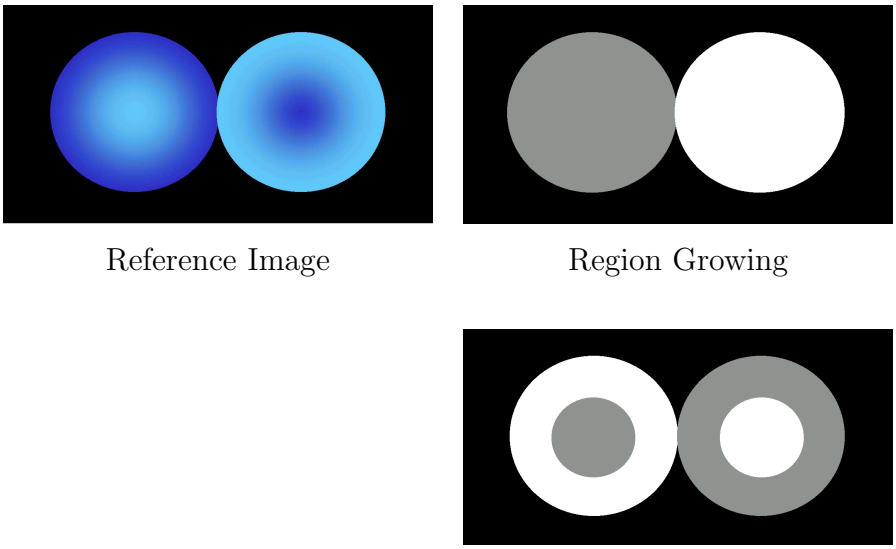


Figure 3.5: An illustration of a clustering method failure. Using a region growing method, the reference image can be correctly segmented since region growing methods usually deal well with slowly varying gradients in image objects. In this case, region spilling is an asset. However, when applying clustering (irrespective of how many clusters are chosen), the method will fail since spatial information is not being taken into account.

	Clustering	**Spatial**	**Energy Minimization**
Computational complexity	Excellent	Good	Poor
Implementation ease	Excellent	Excellent	Excellent
Compactness in feature space	Excellent	Poor	Excellent
Spatial compactness	Poor	Excellent	Excellent
Criterion flexibility	Poor	Poor	Excellent

Table 3.1: Comparison of attributes for segmentation algorithms. The computational complexity of single-site energy methods is poor and as a results accelerated methods are necessary in order for energy methods to approach the complexity of spatial and clustering methods.

segmentation problem, and how to find its optimal solution. The use of local (possibly global) contextual constraints is indispensable for reliable and effective image segmentation. It allows energy minimization methods to offer a framework capable of capturing important aspects of the problem being solved. Indeed, a wide variety of constraints can be proposed: clustering in color or other feature spaces as in Section 3.2, spatial similarity as in Section 3.3, region size [142], textures [22], edges [48], etc. Gibbs Random Fields [48, 157, 88] provide a way of modelling these constraints. In particular, given some stochastic constraints of the form [41]

$$\left(\sum_j \underline{l}_{i,j}^{\ T} \underline{x}_j \right)^2 = \mathcal{N}(\mu_i, \sigma_i^2) \tag{3.3}$$

implying that some squared function of feature elements is distributed about an expected value μ_i with some expected variability, then these constraints imply a Gibbs distribution on $\{\underline{x}_n\}$ given by

$$p\left(\{\underline{x}_n\}\right) = \frac{1}{Z} \prod_i e^{-\left\{ \left(\frac{\sum_j \underline{l}_{i,j}^{\ T} \underline{x}_j}{\sigma_i} \right) \right\}^2} \tag{3.4}$$

It is computationally most efficient if the constraints are local, involving only few, proximate pixels, in which case the model is equivalent to an MRF; however, non-local constraints on region size are permitted and do not necessarily nullify the Markovianity property.

The family of piecewise constant models such as the Potts/Ising models (2.17) are very suitable for image segmentation problems [11] and will be used extensively in this thesis. Their main characteristic is the modelling of large constant-label pixel groupings which is very useful in image segmentation where images are composed of patches of homogenous features (*e.g.*, black car, green tree, blue sky, etc.). Therefore, the appearance of large constant-label patches as a solution is an important assumption in the design of our energy minimization framework.

Finding the optimal solution is often challenging due to the non-convex nature of the objective function. Several optimization approaches for computing the local or global optimum have already been discussed in Chapter 2 such as simulated annealing [19, 76], iterated conditional modes [7] and highest confidence first [25]. However, the computational

complexity of the global optimization algorithms such as simulated annealing is usually very high [48]. There are also sampling issues associated with the use of appropriate statistical samplers (Gibbs Sampling [48], importance sampling [33] or other variants).

3.4.1 Ising/Potts Models

Markov Random Field methods are rooted in the seminal work of Geman and Geman [48] who formulated a procedure based on the Potts (2.17) model for noise filtering and image segmentation tasks. MRF methods have been mostly used for parameter estimation of textures [10, 21, 22, 98, 104, 106] which are later classified using other algorithms. They have also been used as image segmentation models to actually drive the segmentation process [48]. The former has been studied extensively in the literature for grayscale textures [21, 22, 104], and color textures [98, 106]. The latter has been first studied in the context of noise filtering [48] and was later used as a pixel grouping algorithm in image segmentation [48, 163]. In [107, 163], the authors introduce methods initialized by vector quantization or clustering algorithms followed by an MRF-based procedure to locally refine the image segmentation result. This is done using a modified k-means algorithm and followed by a Euclidean-distance based MRF model for grayscale images [107] and multispectral images [163].

Gaussian Markov Random Field-based (GMRF) models were one of the first MRF-related tools used for image segmentation [21]. This involved deriving parameters for each texture using a GMRF model and then proceeding with the segmentation process as a separate hypothesis-testing procedure applied to the GMRF likelihood. The most crucial part of this type of procedure lies in determining the actual model parameters. This can be done by using a fuzzy logic-based clustering algorithm [104] or by iteratively estimating model parameters from the segmentation results [21, 22]. For color image segmentation the GMRF defines a spatial texture with explicit dependencies between the R, G, B components [106].

Several early MRF approaches exist [31, 67, 91, 127, 158]. Daily [31] studies an MRF model based on a normalized color metric [61] and line processes. Daily proposes three different neighborhood structures: rectangular (used most often), hexagonal and triangular. He segments images by introducing or deleting line processes thereby merging or

separating individual pixels. Sung [127] uses a MRF model with an angular color difference measure with objective function components enforcing "closeness" to the original image, smooth color transition between neighboring pixels and a line process term which forces a boundary between pixels. Wright describes a method based on Gibbs Random Fields and line processes [158].

Several MRF/GRF methods have been devised for color image segmentation in recent years [46, 60, 73, 86, 101, 163]. Most of these methods use similar prototype-driven algorithms [107]. Yamazaki and Gingras [163] propose a method to refine k-means clustering results using post-processing with an MRF. Gao *et al.* [46] describe a second order MRF coupled with an Expectation-Minimization (EM) algorithm to segment color images in CIE *Luv*.

Mukherjee [101], on the other hand, first segments the color image using a region growing approach and refines the result using an MRF. Kato *et al.* uses a similar approach also in the CIE *Luv* space using a coarse segmentation to drive a first order prototype-driven MRF [73]. Hazel [60] proposes a clustering MRF method coupled with a Gaussian MRF texture model for general multispectral images.

3.4.2 Region Prototype Models

We can similarly construct GRF models with global constraints, such as cluster means, along the lines of Section 3.2. The GRF energy now contains a term penalizing the distance between a pixel and its associated prototype [107]:

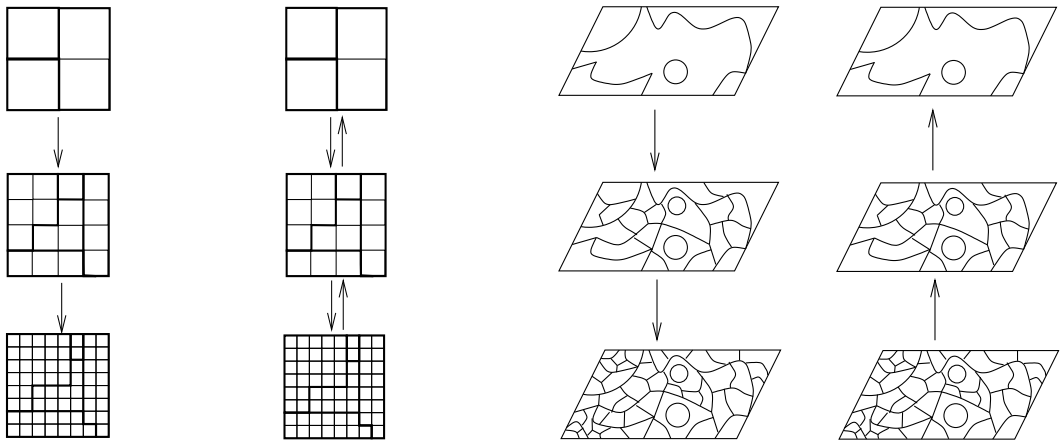
$$U[\{l_{i,j}, \underline{w}_k\}] = \sum_{i,j} \quad \alpha\Phi(\underline{x}_{i,j}, \underline{w}_{l_{i,j}}) + \beta\left[(1-\delta_{l_{i,j},l_{i+1,j}}) + (1-\delta_{l_{i,j},l_{i,j+1}})\right] \tag{3.5}$$

where $\{\underline{w}_k\}$ represent the cluster prototypes. The prototypes $\{\underline{w}_k\}$ may be treated as deterministic, precomputed by another method such as k-means, or they may be stochastic and themselves estimated, jointly with region labels, by the GRF optimizer. Whereas the standard k-means algorithm only involves the first criterion of (3.5), that of pixel similarity, here we have additional flexibility.

Several prototype-based GRFs have been developed following Chang *et al.*'s [20] adaptation of Pappas' [107] original algorithm for color image segmentation by modelling the components of the color vectors as independent random variables. A multivariate Gaussian with a space-varying mean function was the class conditional probability model for the image [20, 107]. The means were initialized to the k-means cluster centers. The algorithm alternates between the MAP estimation of the labels and the determination of the class means. The means are obtained over sliding windows whose size progressively decreases thus progressively adapting to the local characteristics of each region. Huang *et al.* [67], and Liu and Yang [91] use scale space filters for preliminary clustering and refine their region boundaries using various types of MRF's.

Comaniciu and Meer [26] used mean-shift clustering, a method which estimates in a non-parametric fashion the modes of the underlying density (*e.g.*, color image histogram). Comaniciu further improved the method by measuring the significance of each cluster using test statistics that compare the estimated density of the cluster mode with the estimated density on the cluster boundary [27]. This leads to the suggestion that saddle points lying on the cluster borders in the spatial domain define the cluster boundary in the feature domain.

Significant work has been done by Zhu *et al.* based on Bayesian conditional probabilities [140, 166, 167]. Originally, Zhu *et al.* [166] proposed the region competition algorithm which is a formalism unifying region growing [59, 92, 137], snakes [71], and the energy/Bayes/MDL criterion [84]. This algorithm is robust in the sense that new region prototypes can be added when needed and the region means adapt to the underlying image (*i.e.*, no special initialization is necessary as in many MRF methods used to refine clustering results [40, 73, 163]). In [140], the authors present the Data-Driven Markov Chain Monte Carlo paradigm for image segmentation in a Bayesian framework. This work attempts to provide a unifying framework for image segmentation by combining edge detection, clustering, region growing, split-merge, snake/balloon, and region competition by showing how each of these realizes Markov chain dynamics. One key advantage of this algorithm is its ability to achieve a nearly global optimal solution independent of initial labelling. However, it is still not much faster than a flat field annealer.



Top-Down Regular Hierarchy	Multi-Grid Monte Carlo	Top-Down Irregular Hierarchy	Bottom-Up Irregular Hierarchy

Figure 3.6: Illustration of various accelerated annealing estimation methods with their qualitative speeds and other attributes of interest listed in parentheses: top-down regular hierarchies (slow, blocky regions), multi-grid/multi-resolution (fast, continuous data), top-down regular hierarchies or graph cuts (fast, arbitrary regions), and bottom-up irregular hierarchies (fast, arbitrary regions). The arrows indicate the direction of the processing. The top row represents the coarsest level of processing while the bottom row represents the finest level.

3.5 Gibbs Sampling Acceleration Methods

The major impediment to using energy minimization methods is the significant computational complexity of Gibbs Sampling using global optimization methods such as simulated annealing. A number of methods have been developed in order to speed-up this process. They include top-down regular hierarchies [72], top-down irregular hierarchies also known as graph cuts [11, 124, 160], bottom-up irregular hierarchies [2, 17, 99] and cluster sampling [3, 4]. These different paradigms are illustrated in Figure 3.6.

3.5.1 Top-Down Regular Hierarchies

There are several ways one could envision speeding up MCMC algorithms. One method would be to create top-down hierarchies which refine their parameters at ever finer levels [72]. These top-down methods are problematic since they force a square partition on all scales and, therefore, the graph partitioning results on image segmentation are "blocky."

Multi-grid Monte Carlo (MGMC) [55] provides a formulation for interaction between various components of the energy function from those on a coarser scale to those on the finest scale. In MGMC, estimates computed at different data granularities or hierarchies influence one-another through top-down and bottom-up feedback. MGMC has not been applied to discrete labelling problems and, therefore, it is not possible to comment on its performance in this thesis.

3.5.2 Top-Down Irregular Hierarchies or Graph Cuts

A graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ can be partitioned or *cut* into two mutually exclusive sets V_A and V_B such that $V_A \cup V_B = \mathcal{V}$ and $V_A \cap V_B = \varnothing$ by removing all the edges connecting V_A and V_B. The total weight of the removed edges provide the degree of dissimilarity between these two parts. The graph cut is then defined as [124, 160]

$$cut(V_A, V_B) = \sum_{v_i \in V_A, v_j \in V_B} (v_i, v_j) \tag{3.6}$$

where (v_i, v_j) is the edge weight between a node (or pixel) in V_A and a node in V_B usually computed using $\Phi(i, j)$. The optimal partitioning of this graph minimizes $cut(V_A, V_B)$ which is equivalent to finding the maximum flow from the V_A to V_B [11]. Each of the subgraphs V_A and V_B is then subdivided in turn until no more partitioning is required with respect to some criterion [160]. In this way, graph cuts can be viewed as a top-down irregular hierarchy.

(3.6) is a globally optimal criterion which favors cutting small sets of isolated nodes in the graph since it increases with the number of edges going across the two partitioned parts. Thus, the method has a built-in bias for partitioning out small sets of points. In order to avoid this disadvantage, Shi and Malik [124] developed the normalized cut criterion to partition the graph based on both the total dissimilarity between the different groups,

as well as the total similarity within the groups. Instead of looking at total edge weight connecting two potential partitions, their criterion computes the cut cost as a fraction of the total edge connections to all the nodes in the graph. Another method consists of generating an "average" cut in a similar graph based on many sample cuts [47]. The stochastic nature of their method makes it robust against noise, including accidental edges and small spurious clusters. Normalized cuts presents many of the problems of clustering algorithms such as the representation of global spatial clusters in images.

Graph cut methods also include minimum-cut [118] and graph-cut [11] which formulate energy minimization problems in a maximum flow framework in order to solve them in polynomial time, and generalized belief propagation on graphs [164]. These different graph partition algorithms are all based on adjacency graphs encoding local feature interactions and have difficulties to scale up to global spatial interactions which occur in images. Furthermore, the transformation of an energy minimization problem into a maximum flow problem might not be not possible in many cases which is potentially problematic for graph-cut and minimum-cut. Finally, because it can only be applied to specific energy forms, generalized belief propagation is inadequate and, therefore, not applicable in general.

In nested graph cuts, [142] creates a top-down hierarchy of regions by starting with a few large segments or cuts at the beginning and refining those cuts until no more graph cuts can be made. The presented hierarchy has a stopping criterion in the form of a threshold on component size (which eliminates the possibility of detecting small objects). Otherwise, the segmentation would produce a highly oversegmented result as many small regions would remain.

Graph cuts is a top-down framework for image segmentation that achieves excellent results as shown in the reviewed research. Irregular bottom-up hierarchical schemes are reviewed next.

3.5.3 Bottom-Up Irregular Hierarchies

The notion of irregular pyramids or tessellations [99] most closely resembles our framework in structure with significant differences in the method details. Irregular pyramids provide a bottom-up framework for gradually building regions in an image based on sampling the data at lower levels in the hierarchy. The method explicitly uses a stochastic process to

decimate the number of nodes in a graph. This reduction factor is at least four due to the condition that no two neighbors may survive. This is a limiting condition for this method as reduction factors for other methods based on irregular pyramids may be much larger. Furthermore, the graph is ultimately reduced to one node. This methodology has several shortcomings. First, since no two adjacent vertices may survive to the next level, small image regions could easily be eliminated. Second, a stoping or fitness criterion is needed to determine which level of the hierarchy provides the best segmentation which is an ambiguous problem formulation. Finally, a representative pixel of the region is kept at the next level and used for edge weight computation which may not be representative of the region as a whole.

Several other methods related to irregular bottom-up pyramids have been developed in recent years and exist under different names; namely, hierarchical watershed-type algorithms such as repeated waterfall [2], hierarchical watersheds [13], combinatorial pyramids [12] and highest confidence first [25]. All algorithms have greedy or deterministic optimization schemes.

The repeated waterfall algorithm [2] applies repeatedly a watershed-like algorithm to segment images. In [13], Brun *et al.* apply the watershed algorithm in a bottom-up framework building a hierarchy of regions until all regions are merged. Both algorithms need an experimentally devised stopping criterion in order to stop or some other way to pick the desired segmentation result. Both methods are ad-hoc and do not use a principled framework to decide on the segmentation result.

Chou and Brown describe the Highest Confidence First algorithm [25] where the grouping resulting in the largest total energy decrease is chosen (cf. see Section 2.7 for more details). They effectively create a hierarchy of regions which differ by one in number from one level to the next. This method is prone to getting stuck in a local minimum, however it is computationally efficient at least for small graphs [94].

These algorithms first create region groupings based on some features and then merge those regions (effectively the nodes in our graph) together. The very nature of image processing dictates that local pixel-based features are usually very different from global region-defining features. This naturally leads to the creation of two models: one for defining a region and one for the merging of regions. However, in a bottom-up accelerator, the

energy at the coarsest scale and the finest one should be related if not the same. In all of these methods, a relationship between levels exists, however, the energy being minimized at the coarsest scale does not correspond to the energy at the finest scale.

A related bottom-up hierarchy is presented in [17]. This method is a computationally efficient hierarchical bottom-up algorithm based on a "divide-and-conquer" approach to network flow for phase unwrapping. In this method, the image is subdivided into rectangular or square blocks where the phase is unwrapped independently of other blocks effectively creating many small phase unwrapping subproblems. These blocks are recombined to produce the unwrapped phase at the next higher level creating another unwrapping problem (which is now easier since the phases within the blocks have been unwrapped and the blocs themselves are now being unwrapped with respect to each other). The unwrapped phase in the original blocks has an irregular structure that then grows as the unwrapped phases are combined at higher levels in the hierarchy.

The reduction in computational complexity depends on the size of the initial blocks the image/graph is subdivided into. The blocks cannot be *too small* (since this would mean that the original partitioning problem might not be addressed) nor could they be *too big* (due to increased computational complexity) and therefore some intermediary size needs to be chosen. In the case of Carballo [17], a block size of 100×100 is chosen at the finest level. This hierarchy reduces the computational complexity by a factor of approximately 10; however, it allows images of virtually infinite size (limited by the computer's hard disk space) to be processed. However, this computational complexity improvement might not be adequate for large images.

3.5.4 Cluster Sampling

To accelerate discrete estimation via MCMC one must look at computations on subgraphs, regions or groups of vertices. The Swendsen-Wang (SW) algorithm [128], a method used extensively in computational physics, reduces the computational complexity of the Gibbs sampler [48]. Specifically, SW splits and regroups subgraphs dynamically; however, in its original form it is only applicable to Potts models. SW was recently generalized to sample arbitrary posterior probabilities on graphs [4]. By graph clustering and relabelling this algorithm realizes the splitting, and regrouping of a subgraph, in contrast to flipping a

single vertex via the Gibbs sampler [48].

The improved cluster sampling method simulates ergodic and reversible Markov chain jumps in the entire solution space and is applicable to arbitrary posterior probabilities (or energy functions) defined on graphs. Three variants exist: SWC-1, samples all edges in a current subgraph, SWC-2 starts from a single vertex and grows into a subgraph, and SWC-3 is a SW-based Gibbs sampler. SWC-1 and SWC-2 provide an improvement of two orders of magnitude over the single site Gibbs sampler whereas SWC-3 is slower as it has a higher overhead for each site visit [4]. Barbu and Zhu also present a two-level hierarchic Swendsen-Wang algorithm [3] which carries out graph partitioning on two levels of granularity: the first as in the original algorithm and a second coarser level.

The Swendsen-Wang algorithm is based on groups of nodes and therefore provides a considerable (up to 400 times) [4] computational gain over the single site Gibbs sampler. A more significant speedup is needed if graph partitioning using simulated annealing or ICM is to be used for practical applications on very large graphs such as images. Furthermore, the optimization in [4] is done on preprocessed atomic regions which create small graphs of 500-2000 nodes. The preprocessing steps can introduce a considerable amount of error.

3.6 Discussion and Conclusions

We have surveyed the state of the art in color image segmentation by reviewing three fundamental segmentation contexts: clustering, spatial methods, and energy minimization. In most cases, a baseline algorithm will be straightforward to implement or available in the public domain; however, the quality of the image segmentation will depend greatly on the selected parameters, subtleties in the distance measure, as well as initialization issues and, in the case of color segmentation, color space selection.

Since probabilistic segmentation models such as MRFs can effectively integrate any type of local or global constraint (although some might be difficult to encode), they are intrinsically more flexible than clustering or spatial methods. Computational complexity being the primary drawback of such methods, a new bottom-up hierarchical method has been devised in this thesis and will be discussed in Chapter 5.

Chapter 4

Background: Review of Color Pixel Comparison

The second component of an image segmentation or graph partition algorithm defines the between-pixel or between-node distance measure. This computation is done to ascertain differences between the data. Without a reliable distance measure, it is not possible to determine which nodes should go together and which constitute disparate parts. The appropriate distance measure needs to reflect the kind of problem that is being solved and thus knowledge about the problem is encoded in the distance measure either implicitly (by carrying out comparisons in a feature space assumed to be Euclidean) or explicitly (carrying out comparisons in the sensor space which is not necessarily Euclidean).

Pixel distance measures are computed between adjacent pixels or image patches. In this context, we will limit our examination to color distance measures. We shall distinguish between distance measures (or semi-metrics) and metrics. Φ is called a metric on $\mathbb{R}^d$ (where d defines the dimensionality of the feature vector) if it satisfies the following conditions:

1. Positivity: $\Phi(\underline{x}, \underline{y}) \geq 0$
2. Zero property: $\Phi(\underline{x}, \underline{y}) = 0$ if and only if $\underline{x} = \underline{y}$
3. Commutativity: $\Phi(\underline{x}, \underline{y}) = \Phi(\underline{y}, \underline{x})$
4. Triangle inequality: $\Phi(\underline{x}, \underline{y}) \leq \Phi(\underline{x}, \underline{z}) + \Phi(\underline{z}, \underline{y})$

where $\{\underline{x}, \underline{y}, \underline{z}\} \in I\!R^d$. Φ will be called a semi-metric or a distance measure if it can only satisfy the first three conditions. When a distance measure is a metric, the triangle inequality (condition 4) ensures that distances produced can be ordered. This matters when computing the statistics of the measured distances. However, we often care only whether a difference between values is *small* or *large* when pixel features are close or far away respectively. This type of information can still be obtained using a semi-metric.

Choosing an appropriate pixel distance measure for a particular application can determine whether the algorithm devised to solve the problem will be successful [35]. In the case of color image segmentation, the choice of distance measure will depend on the color space and color model being used to solve the given problem. In this chapter, we will explore physics- or reflectance-based color spaces and color distance measures. The literature will be reviewed and the state of the art described.

4.1 Color Spaces

We need to decide the color space in which the metric or distance measure will be applied. The use of color information has been studied extensively in image processing [68, 145]. A color space is a method by which we can specify and represent color, normally as a three-dimensional vector. The trichromatic model of color representation, commonly known as RGB, is fundamental to the human perception of color [161]. It is based on the additive primary colors red, green and blue [145], and corresponds most closely to human physical sensors for colored light (*i.e.*, the cones in the human eye) implemented as red, green, and blue filters in most color charged coupled device (CCD) sensors.

Even though CCD image sensors have a linear response [129], sensor amplifiers may cause the output image to contain nonlinearities. Furthermore, in the case of images meant for display, an intentional nonlinearity or correction factor is introduced in order for humans to accurately perceive the displayed image on common display technologies, such as a cathode ray tube. The γ function (also called the transducer function) describes the mapping from the CCD values to final image intensity. Its general form follows $A \cdot z^{\gamma}$ where z is the CCD value, A and γ are device-dependent constants [129]. Subsequently, to infer properly CCD signal levels from a final image, we must apply the inverse of the

transducer function to each image pixel in the final image. However, we will assume that we do not need to correct the CCD signal for the γ function.

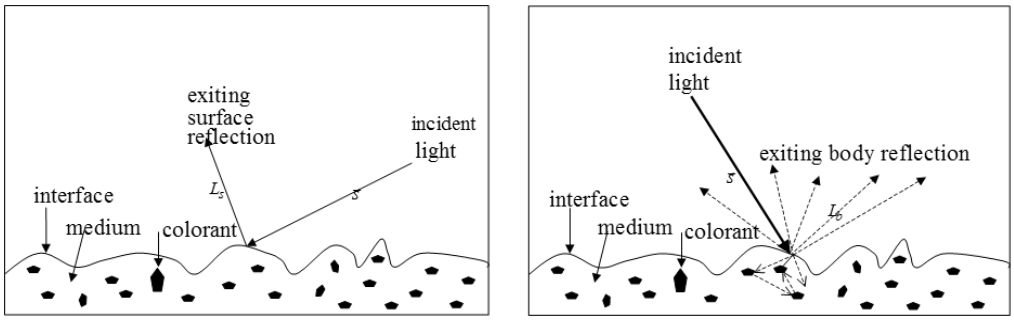
Color can be represented in a variety of interchangeable color spaces [83]. Since RGB is the physical sensor-based color space, all other color spaces are derived from it. Because the vector representation of color is space-dependent, clearly the separation between colors will also be space dependent. Since the three components of the RGB space are highly correlated, RGB pixels are usually transformed into another color space such as CIE Luv [68], CIE Lab [68], Hue Saturation Intensity (HSI) [54, 68, 95], or others where correlation has been reduced.

Human notions of color closely follow the HSI representation; however, this space and others like it are not actually perceptually uniform – *i.e.*, human perceptual color dissimilarity is not proportional to the Euclidean distance between two colors [161]. CIE Luv and CIE Lab are approximately isotropic spaces that were designed to be perceptually accurate with respect to the use of the Euclidean distance metric. Color spaces provide the vector coordinates of individual colors for the image segmentation application. However, for many applications it is not only necessary to operate in some color space, but also to determine whether a particular color representation can be invariant to various illumination effects and to noise, based on a physical and/or probabilistic model of color. None of the above spaces are appropriate for obtaining a physics-based segmentation without some space-dependent preprocessing.

4.2 Physics-Based Reflection Models and Spaces

Color models allow us to make certain assertions, regarding color generation and perception, about the color space that we would like to use based on the laws of physics (optics). Physics-based color models explain how light is reflected from objects in a scene based on the physical properties of materials. They are used when algorithms need to achieve color constancy – the perception of objects in the real world without illumination effects – such as is the case for humans.

Much work has been done on physics-based color modelling over the years [123, 132, 133, 135, 145]. A commonly used physics-based model is Shafer's Dichromatic Reflection



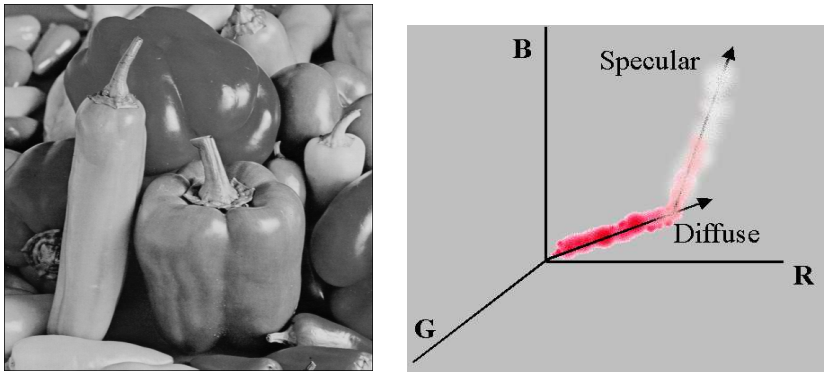
(a) (b)

Figure 4.1: The Dichromatic Reflection Model: (a) specular reflection and (b) body reflection.

Model (DRM) [123, 145], which assumes that light reflected from objects can be separated into specular reflection and diffuse reflection as shown in Figure 4.1. Specular reflection or a highlight is characterized visually by a glossy appearance and describes light that is reflected in a mirror-like fashion from a surface. Diffuse or body reflection is the light reflected in all directions from a surface, giving a surface its colored appearance. Figure 4.2 illustrates the difference between the two reflections. By using the DRM, pixel difference computations can be done directly in the RGB space.

A unichromatic version of DRM has been introduced by Healey [62] and DRM has been described for a variety of materials [135]. Maxwell and Shafer [96, 97] describe a modification to take into account piecewise uniform dielectric objects by formulating hypotheses based on shape, illumination and material properties. Zhu and Yuille propose a modification of the DRM that is not as sensitive to noise and suggest a new method for highlight detection [166]. Other physical models include Phong's shading model [66, 110], and Nayar's hybrid reflectance model [103]. Buluswar and Draper provide an overview of these models, as well as an adaptation of the DRM to be used with a new daylight color model [14]. Finally, atmospheric effect models for outdoor vision are considered in [102]. The focus here will be on inhomogeneous dielectric materials such as plastics and painted surfaces. We will base our analysis on the work by Tominaga [132, 133].

We will now describe the Dichromatic Reflection Model. Light reflected from an object surface o (called the color signal) is described as a function $c^o(\lambda, i, j)$ dependent on



(a) (b)

Figure 4.2: An illustration of specular and diffuse reflections: (a) a sample color image with secularities; (b) the RGB distribution of the big red pepper.

wavelength λ and pixel location $\{i, j\}$:

$$c^o(\lambda, i, j) = \text{Body Reflection} + \text{Interface Reflection} \quad (4.1)$$

$$= \nu(i, j)s^o(\lambda)e(\lambda) + \eta(i, j)e(\lambda) \quad (4.2)$$

where $e(\lambda)$ is the spectral power distribution of the light source, $s^o(\lambda)$ is the spectral-surface reflectance of object o, $\nu(i, j)$ is the shading factor and $\eta(i, j)$ is a scalar factor for the specular reflection term. The following set of equations can then represent the sensor responses for a camera using R, G, and B coordinates:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} (i, j) = \int c^o(\lambda, i, j) \begin{bmatrix} \mathcal{R}_R(\lambda) \\ \mathcal{R}_G(\lambda) \\ \mathcal{R}_B(\lambda) \end{bmatrix} d\lambda \quad (4.3)$$

where $\mathcal{R}_k(\lambda)$ ($k \in \{R, G, B\}$) are the spectral sensitivity functions of the camera in the

visible spectrum. Substituting (4.2) into (4.3), we have

$$\begin{aligned}\begin{bmatrix} R \\ G \\ B \end{bmatrix}(i,j) &= \nu(i,j)\int s^o(\lambda,i,j)e(\lambda)\begin{bmatrix} \mathcal{R}_R(\lambda) \\ \mathcal{R}_G(\lambda) \\ \mathcal{R}_B(\lambda) \end{bmatrix} d\lambda + \\ &\quad \eta(i,j)\int e(\lambda)\begin{bmatrix} \mathcal{R}_R(\lambda) \\ \mathcal{R}_G(\lambda) \\ \mathcal{R}_B(\lambda) \end{bmatrix} d\lambda \qquad (4.4) \\ &= \nu(i,j)\underline{c}_b(i,j) + \beta(i,j)\underline{c}_i(i,j) \qquad (4.5)\end{aligned}$$

where $\underline{c}_b(i,j)$ is the body color vector and $\underline{c}_i(i,j)$ is the illumination color vector. These color vectors are normalized into a unit vector length.

For the sensor outputs R, G, and B to be white balanced, it is necessary to satisfy the following condition:

$$\begin{aligned}\int e(\lambda)\mathcal{R}_R(\lambda)d\lambda &= \int e(\lambda)\mathcal{R}_G(\lambda)d\lambda \qquad (4.6) \\ &= \int e(\lambda)\mathcal{R}_B(\lambda)d\lambda \qquad (4.7)\end{aligned}$$

As long as the illuminant $e(\lambda)$ is a constant white over the visible wavelengths, and the spectral sensitivity functions $\mathcal{R}_k(\lambda)$ ($k \in \{R,G,B\}$) have the same area, then the above condition obviously holds. However, if the illuminant is not white, a color balancing step [68] is needed where the three sensor outputs are adjusted to be equal in power. In this thesis, it will be assumed that the illumination light is white or that the image has been white balanced.

4.2.1 Physics-Based Color Spaces

Some color spaces have been designed using physics-based models of color: normalized color or rgb [61, 62], $c_1c_2c_3$ [125], shading and highlight invariant $l_1l_2l_3$ space [125], highlight invariant spaces $h_1h_2h_3$ [153]. In these spaces, distance is usually computed using the Euclidean distance except in [153] where the vector angle is used.

The rgb space has been used in the literature for several decades [50, 61, 62, 106]. rgb is obtained by dividing the RGB pixel elements by the pixel magnitude. For matte

objects, the color representation in the rgb space is invariant with respect to illumination direction and intensity, as well as the viewing direction and surface orientation [49]. It is still sensitive to specular reflections, and inter-reflection. It is also not well defined for pixels with low intensives.

The measured colors of a region with a uniform color are on the triangular color plane in the RGB space spanned by the body and surface reflection components [49]. Therefore, any expression defining colors on the same linear triangular plane will have similar properties to hue. The $l_1 l_2 l_3$ space was introduced to uniquely determine the direction of the triangular color in the RGB space [49]. Applying the dichromatic reflection theory, one may observe that this space is invariant to highlights, viewing direction, surface orientation, as well as to illumination direction and intensity.

4.2.2 Probability-Based Color Reflection Models

Probabilistic color models were devised to take into account noise in images. The noise is most commonly modelled using a normal distribution [166] although Sung [127] models color noise using a Rayleigh distribution. Zhu and Yuille propose a model which separates light into three components [166]

$$\begin{aligned} c^o(\lambda, i, j) &= \text{Body Reflection} + \text{Interface Reflection} + \text{Noise} && (4.8) \\ &= \nu(i,j)c^o(\lambda)e(\lambda) + \eta(i,j)e(\lambda) + \upsilon(i,j) && (4.9) \end{aligned}$$

where $\upsilon(i, j)$ represents the noise or residuals that are Gaussian distributed. The authors propose a series of operations for highlight or specularity detection. Residuals for body and specular reflections within a region have sufficiently different distributions to distinguish between them.

4.3 Distance Measures

The key to color image segmentation is to apply the appropriate color distance measure for the problem at hand. The choice of distance measure can greatly affect image segmentation or clustering results [35]; therefore, it is critical to make sure that the similarity measure

being used is the appropriate for the assumed color space. Several distance measures are summarized below.

4.3.1 Euclidean Distance

The most often used distance measure due to its mathematical properties and ease of use is the Euclidean distance. Recall its definition in (1.2) [35] given here for clarity of presentation:

$$\Phi_E(i,j) = (\underline{x}_i - \underline{x}_j)^T(\underline{x}_i - \underline{x}_j). \tag{4.10}$$

However in the case of color images, where each pixel is represented as a RGB vector, the Euclidean distance is a particularly poor measure of color similarity because the RGB space is *an*-isotropic, especially when lighting effects such as specular reflection and shading are present in the image. Consider the image in Figure 4.3 where the green (or brown) colors are closer to each other perceptually than in terms of the Euclidean distance. It is apparent the Euclidean distance between dark green and dark brown (or similarly light green and light brown) is small due to their intensity.

On the other hand, in the CIE *Lab* and CIE *Luv* spaces, the Euclidean distance represents approximately the color difference as perceived by humans [122].

The mean of a set of pixels compared with the Euclidean distance can be obtained using the vector mean of those pixels. For grayscale images, this is just the mean of those values while for color or other multispectral pixels the mean is computed for each color or spectral band individually with those values being then aggregated in a vector.

4.3.2 Mahalanobis Distance

The Euclidean distance can be generalized to the Mahalanobis (or general weighted Euclidean) distance [62, 166]:

$$\Phi_M(\underline{x}, \overline{\underline{x}}) = (\underline{x} - \overline{\underline{x}})^T\Sigma^{-1}(\underline{x} - \overline{\underline{x}}) \tag{4.11}$$

where Σ represents the covariance matrix for vectors $\{\underline{x}\}$ with a mean $\overline{\underline{x}}$. The Mahalanobis distance is a probabilistic generalization of the Euclidean distance based on the Gaussian probability distribution. We will come back to probabilistic distance measures in Chapter 6.

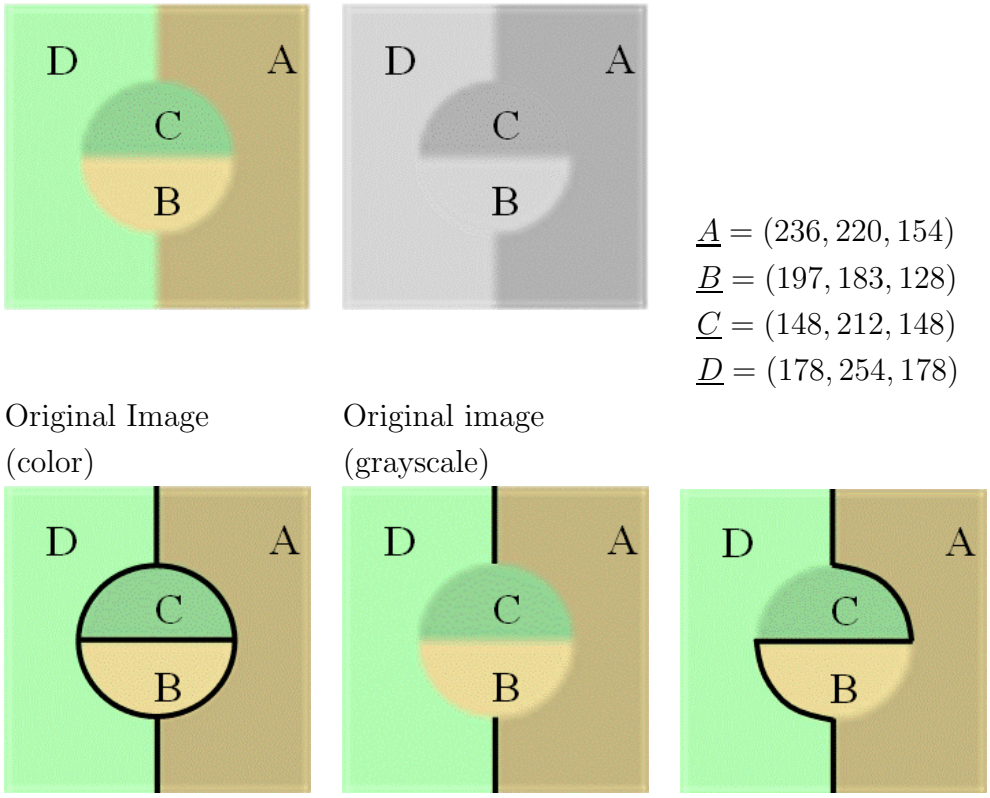


Figure 4.3: Color segmentation using different distance measures. The Euclidean distance cannot properly distinguish between the two regions of green and brown pixels using any threshold. On the other hand, vector angle is able to separate both regions exactly. Th represents the thresholds applied to the different distance measures.

The mean of a set of pixels compared with the Mahalanobis distance can be obtained using a weighted vector mean of those pixels. For grayscale images, this is just the weighted mean of the values of a set of pixels while for color or other multispectral pixels the weighted mean is computed for each color or spectral band individually with those values being then aggregated in a vector.

4.3.3 Vector Angle

The vector angle measure (1.3) or its variants has been used a few times in the literature [127, 148]. The vector angle measure in this form is a semi-metric. Its original definition (1.3) (see page 5) is repeated here:

$$\Phi_V(i,j) = 1 - \left(\frac{\underline{x}_i{}^T \underline{x}_j}{|\underline{x}_i| \cdot |\underline{x}_j|} \right)^2 \tag{4.12}$$

Because the dot product between the vectors is divided by the magnitude of the vectors, vector angle has been shown to be intensity invariant with respect to the Dichromatic Reflection Model [150]. Figure 4.3(e) shows that two colors of different intensities and an angle of 0 would be identical with respect to the vector angle. The main problem with the vector angle is that it gives very "noisy" results for vectors with small magnitudes [127] and is undefined if $|\underline{x}| = 0$ or $|\underline{y}| = 0$.

The "mean" of a set of pixels compared with the vector angle distance is an indication of the general direction of the pixels in that set. Here we no longer care about the intensity of pixel values and a Euclidean mean; rather, we compute the principal direction of the set. Calculating the principal direction is equivalent to obtaining the principal component vector corresponding to the largest eigenvalue of the covariance matrix of the data [34, 148]. A similar approach is adopted by Zhu [166] where the albedo or body "color" of the region and the average intensity of each pixel is obtained by minimizing the sum of squared error with respect to the original pixel colors over all points within the region. The minimization is carried out using steepest descent although other methods can also be used.

4.3.4 Histogram-Based

For completeness, we discuss the Kullback-Leibler divergence (KLD) [4, 80] used to compute differences between probability distributions. This method is based on region histograms as opposed to feature values like the previous distance measures. Barbu and Zhu [4] use a Canny edge detector [15] to detect edges which is followed by an edge linking heuristic in order to obtain atomic regions. The histograms of the regions delineated by Canny edges are then determined and the differences between these histograms are calculated using KLD [4, 80]:

$$\Phi_{KLD}(i,j) = e^{-\frac{1}{2}(KLD(h_i||h_j)+KLD(h_j||h_i))} \tag{4.13}$$

where KLD is the Kullback-Leibler divergence function, i and j are region indexes, and h_i is the histogram of region i. This distance formulation is very useful especially when considering non-homogenously colored textured areas where a classical between pixel distance measure cannot be used. However, in order to use this measure one has to preprocess the image using edge detection and edge linking to create some "homogenous" regions where the histogram can be computed. This results in a problem formulation which is no longer contingent on a single model and makes the analysis of the results more difficult (*i.e.*, the reliability of results in such a scheme would depend a great deal on the accuracy of the edge detector and not the pixel grouping mechanism). It would be possible to apply KLD to pixels by computing the histogram based on some window around the given pixels. However, produced histograms would be unreliable near the occurrence of edges.

4.4 Discussion

The Euclidean distance has been applied to virtually all color spaces whether or not it was the most appropriate choice. There are many instances where this similarity measure fails such as in images with a lot of variation due to illumination. In many cases, the spaces are not perceptually uniform (unlike CIE $L^*a^*b^*$ and CIE $L^*u^*v^*$) and, therefore, the Euclidean distance is a poor measure of color similarity [122, 148]. However, it is appropriate to use this measure if images present little noise. The Mahalanobis distance

generalizes the Euclidean distance to deal with Gaussian-distributed additive noise and has proven to be effective [62, 166].

The vector angle is an appropriate tool when intensity invariance is desired in a distance measure. However, it is not useful for segmenting images other than those with non-zero chromaticity values, unless these are somehow taken into account [127].

Defining the appropriate distance measure Φ is a crucial part of the challenge to design a color image segmentation algorithm. It is certain that without an appropriate distance criterion, segmenting images will not be possible irrespective of the sophistication of the grouping algorithm. As a result, we distance new probabilistic measures in Chapter 6.

Chapter 5

Pixel Grouping: Stochastic Nested Aggregation

In this chapter, a major thesis contribution will be described; namely, the speed up of Markov Chain Monte Carlo (MCMC) methods such as Gibbs Sampling using an effective strategy by creating a bottom-up hierarchy based on *stochastic nested aggregation*[1] or SNA. The motivation behind this method is simple. Discrete state estimation or labelling using stochastic optimization techniques such as Simulated Annealing [76] is usually very computationally expensive, on the order of $O(N^3)$, as is illustrated in detail in Section 5.1. Although such computationally expensive algorithms can converge to a local minimum close to a global minimum (convergence to a global optimum is only guaranteed for an effectively exhaustive search of the solution space [48]), the computational complexity makes them impractical.

Stochastic Nested Aggregation has the following general characteristics each of which will be explained in more detail in this chapter:

- SNA is a bottom-up aggregation method where pixels are locally grouped into regions of increasing size resulting in algorithms with an overall computational complexity of $O(N)$.
- SNA is a global optimization framework that can be used for stochastic and deter-

[1] A preliminary treatment of this approach was given in [155, 156].

ministic methods as it helps to avoid many poor local minima.

- SNA is stochastic in nature at each level of the hierarchy and, therefore, moves are reversible within a level (but not between levels).
- Because of the reversibility of moves, SNA is restricted to energy models with pairwise comparisons such as the first order Potts model.
- There exists energy models such that SNA is scale invariant and it can solve exactly the same optimization problem at all levels of the hierarchy.

In this thesis, we are interested using an accelerated framework for stochastic and deterministic optimization methods in order to partition a graph of labels $\mathcal{G}$ (cf. Section 2.1, page 14) composed of nodes v_i and edges (v_i, v_j) where $i, j = \{1, \ldots, N\}$ and i and j are each other's neighbors. The goal of graph partitioning is to obtain a smaller graph by grouping similar nodes together by assigning them the same label. For example, in image processing we would want to group nodes or pixels with the same features to partition or segment an image into objects or other meaningful parts. We partition this graph by minimizing an energy function U defined on the graph. In image segmentation, we are especially interested by the piece-wise constant class of functions which is characterized by large areas of constant label (in image processing terminology) or color (in graph theoretic terminology). Such an energy function can be devised to have a unique global optimum and therefore provides the means of partitioning a graph with respect to a single global criterion that is applicable at the finest and coarsest levels of processing the graph.

One can envision a general technique which first *quickly* gathers like nodes (as defined by the model) into small groups in order to estimate their labels collectively. Then, some node groups can in turn be collected *quickly* into somewhat bigger groups. This cycle continues until no more groups need to be aggregated. We are then creating progressively smaller single site graphs that perform multi-site discrete estimation. The nested aggregation formulation will proceed from the *finest* level (in image segmentation: pixels) through intermediate region groupings to a final region configuration. This concept should also be applicable to *any* problem involving stochastic or deterministic[2] graph partition. Figure 5.1 shows an example of the progression of nested aggregation.

[2]Although we refer to deterministic algorithms in this thesis, these algorithms are not entirely devoid of

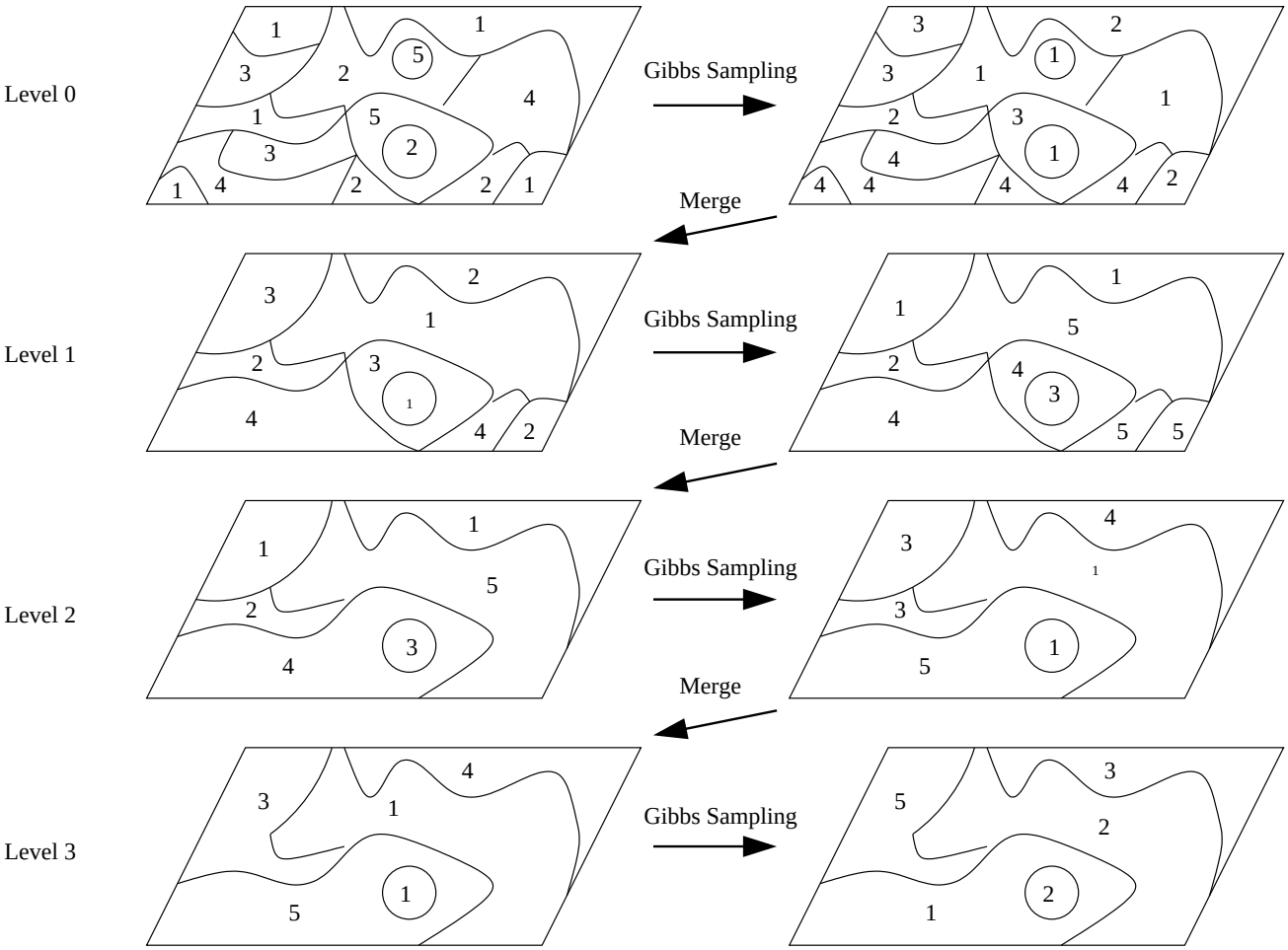


Figure 5.1: Stochastic Nested Aggregation: the algorithm starts with a random label assignment at Level 0 and proceeds to merge regions successively by applying very fast annealing schedules to the Gibbs Sampler at various levels in the hierarchy until no more subgraphs need to be merged. The left column indicates the state of the labels at the start of each level while the column on the right indicates the final label configuration. Between level pairs, adjacent subgraphs with identical labels are merged before the process continues. If no subgraphs need to be compressed into single nodes, then the method is deemed to have converged.

Interestingly for graph partitioning, nested aggregation can make both stochastic and deterministic optimization paradigms more effective. First, even though deterministic methods reach *local minima* very quickly, using nested aggregation the reduced order graph created from the local minimum is no longer a local minimum in the new configuration. This allows nested aggregation to proceed to *much better* local minima since the algorithm usually has more than one level. Second, stochastic methods such as simulated annealing can be run considerably faster while preserving the local minima avoidance property. The issues of faster stochastic optimization and better local minima avoidance by deterministic optimization are related and will be discussed in detail in Section 5.2.

The purpose of this chapter is to describe an original contribution to the literature; namely, the general stochastic nested aggregation framework, as well as the hierarchical generalization of the Potts model as an example of the methodology. By definition, stochastic nested aggregation simulates ergodic and reversible Markov chain jumps in the space of graph partitions at *each* level in the hierarchy as long as the optimization scheme (*e.g.*, simulated annealing) permits it and is only applicable to pairwise energy functions defined on graphs.

This chapter is organized in the following manner. Section 5.1 discusses the Gibbs Sampler and its limitations. Section 5.2 presents the general concept of stochastic nested aggregation which is followed by Section 5.3 with a discussion on the adaptation of the Ising/Potts models to the stochastic nested aggregation framework which also includes preliminary results. A discussion of extensions based on the framework follows in Section 5.4. A final set of comprehensive results is presented in Section 5.5. The chapter concludes with a brief summary.

---

randomness. Namely, a deterministic algorithm could perform gradient descent; however, in flat regions, depending on its implementation, it might engage in a random walk.

5.1 Gibbs Sampler and Its Limitations

The difficulty of sampling in the partition space Ω_π can be illustrated using the Ising model [40, 48, 72, 88, 157]. Consider the following Ising/Potts model

$$U(\ell) = -\beta \sum_{(i,j)\in\mathcal{E}} \delta_{l_i,l_j} \tag{5.1}$$

where δ_{l_i,l_j} is the Kroenecker delta and β is the region coupling parameter that is a non-zero constant. β controls the degree to which we want to create an edge; a larger β means less edges which will become an important concept later in this chapter. The labels $\{l_i\}$ can take on two distinct values: "+1" and "−1." To obtain a solution to this model we use simulated annealing [76]. We can write the Gibbs form (2.11) of (5.1) as

$$p(\ell) = \frac{1}{Z} e^{\left\{-\frac{\beta}{T}\sum_{(i,j)\in\mathcal{E}}\delta_{l_i,l_j}\right\}}. \tag{5.2}$$

We will be using the Potts model for our analysis throughout the thesis contribution chapters as it is a piecewise constant model characterized by large patches of constant or homogenous areas punctuated by sharp discontinuities or edges. We use this model since it represents the image segmentation problem in a believable manner. Indeed, by segmenting an image, we would like to find all the areas in it that are homogenous with respect to some set of features (*e.g.*, color). Finally, the cliques in the Potts model are also pairwise making it suitable for the SNA framework.

5.1.1 1-D Analysis

Without loss of generality consider a 1-D sequence N "−1" spins[3] between two infinite groups of "+1" spins shown in Figure 5.2(a) over a homogeneous data field (*i.e.*, the pairwise data comparisons or differences are exactly zero). Suppose we apply the Ising model (5.1) to this sequence using the Gibbs sampler. The highest probability for (5.2) or lowest energy for (5.1) will be obtained when all spins are facing in the same way (*i.e.*, "+1"). Irrespective of the value for T, the spins defining the boundary between the two

[3]Note that in general the computational complexity will be the same irrespective of the initial spin configuration of the N spins.

domains do not affect the energy if flipped which leads to a random walk on the boundary locations. A random walk is one possible formalization of the idea of taking successive steps (each step being in a random direction), either "-1" or "$+1$."

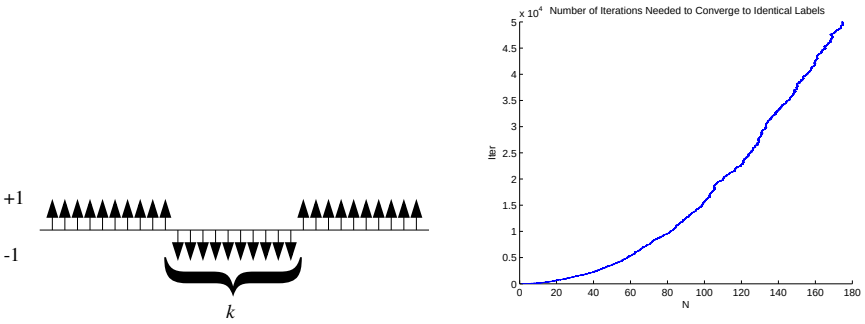
Consider two cases:

- If $T = 0$, then spins within each domain will remain stable and only spins at the boundaries will change creating a random walk of spins at the boundaries.
- If $T > 0$, then spins within each domain can also change with non-zero probability which leads to having a sequence with some of the spins changing in spite of the neighboring spins. Therefore, the walk of annealing has more difficulty in converging to the stationary probability $p(W|X)$. Because we have a non-zero probability of turning a spin into another in the middle of the sequence, we no longer have a pure random walk which leads to more complex behavior and higher convergence complexity.

To simplify convergence speed analysis in 1-D, we will use $T = 0$ in order to obtain a random walk (however, for completeness of analysis we will present image segmentation results for different values of T). Since the "step" size in the random walk is 1 (*i.e.*, the magnitude of the spins), then after N steps the average number of spins which will have the same orientation will be of the order of $\sqrt{N}$. Therefore, to obtain N spins in the same direction, we will need $O(N^2)$ steps or iterations. We devise a simple experiment to show this. Consider Figure 5.2(b) which shows how many iterations it takes to flip a sequence of N spins to be either all "-1" or "$+1$" using a Monte Carlo simulation. Based on the experiment, it is clear that the Gibbs Sampler needs to wait on average $O(N^2)$ iterations for each of N spins to change the label of this sequence. For simulated annealing, this results in an overall $O(N^3)$ complexity for a sequence of N spins.

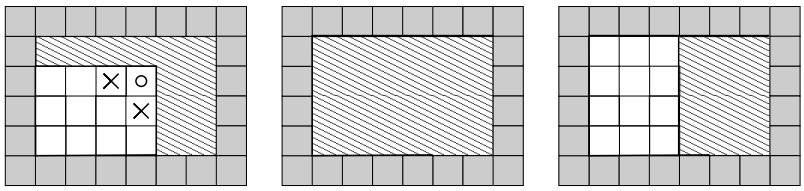
5.1.2 2-D Analysis

In 2-D, the convergence of Gibbs Sampling for an Ising/Potts model (5.1) is further complicated for several reasons. First, given a regular grid in 2-D, the boundary is much more complex with the possibility of moving arbitrarily in the vertical and/or horizontal directions, not just along one axis, which results in in a complex walk of the boundary locations



(a) (b)

Figure 5.2: Illustration of the 1-D slow random walk of annealing. (a) Suppose at $T = 0$ we have a 1-D Ising model with k successive elements inconsistent with the remainder of the domain. The Ising model has some $\beta > 0$. The "-1" and "$+1$" spins that are flipped at the border create a random walk at the boundary since there is no effect on the energy by choosing either a "-1" or "$+1$" and because $T = 0$ prevents the flipping of elements away from the boundary. (b) The plot shows how many iterations it takes to obtain a sequence of N spins of "$+1$" starting with a set of random spins. A Monte Carlo simulation was run for 1000 instances to obtain this curve. It is clear from the figure that the number of expected per spin iterations is approximately $O(N^2)$ giving a total complexity for the algorithm of $O(N^3)$.



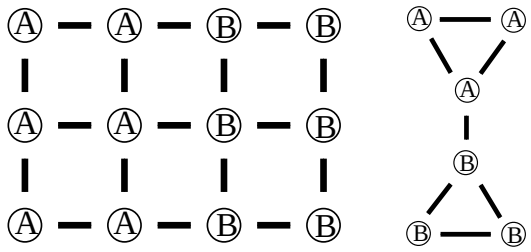
(a) (b) (c)

Figure 5.3: Consider a slow walk of annealing illustration in 2-D between two regions in a homogenous image (*i.e.*, no energy gradient) for a two-label assignment (Ising model) at $T = 0$. (a) Within the domain of flat energies, the annealer performs a walk similar to the 1-D random walk described in Figure 5.2 eventually finding (b) one of the optimal endpoints (in this case a shaded or unshaded region) or (c) a local minimum or deadlock. These are possible endpoints of the walk of annealing in 2-D which the annealer cannot escape. Considering that the probability of flipping each vertex from "+1" to "−1" is $p_o = 1/2$, vertex o has a 50% probability of becoming "+1." If it does become +1, then pixels $\times$ will have the same choices as pixel o and so on.

as is illustrated in Figure 5.3. Furthermore, for an irregular grid (in the case of atomic regions), the boundary movement is based on the structure of the graph.

Second, we no longer refer to the walk of annealing as a *random* walk since there is a possibility in 2-D of reaching a local minimum from which the annealer cannot escape. The convergence to the type of local minimum depends strongly on the structure of the lattice underlying the graph. There are two possible lattices for graph partition: a regular lattice which is a regular structure imposed on the data, and an irregular lattice which adapts to the underlying structure of the data where adjacent subgroups are represented by nodes connected by edges. Examples of these latices are shown in Figure 5.4. If the lattice is regular a local minimum will occur if a horizontal or vertical border develops. Local minima are also possible on an irregular lattice; however, they are structure-dependent due to the arbitrary lattice configuration. Figure 5.4 illustrates the possible deadlocks that can occur for regular and irregular lattices. Deadlocks do occur on irregular lattices; however, their exact form is dependent on the local region of the underlying problem.

The computational complexity of the 2-D walk of labels is difficult to evaluate analytically given the possibility of falling into a local minimum. A test was conducted to assess



(a) (b)

Figure 5.4: Possible deadlocks for piecewise flat models such as the Potts model. (a) If we use a first order (4-pixel) neighborhood on a regular grid, usually applicable at the pixel level, the convergence of an Ising/Potts model will deadlock if a horizontal or vertical boundary occurs (for second order or 8-pixel neighborhoods there would be additional deadlocks on diagonals). (b) On an irregular grid, usually applicable at the image region or patch level, deadlocks can occur, dependent on the geometry of the lattice and the structure of the region neighborhoods.

the *minimum* computational complexity of using Gibbs sampling with simulated annealing to fill a region of constant intensity with the same label using Potts model (5.1). ICM was not used since it is capable of only converging to a local minimum and is prone to getting stuck in a deadlock such as the one in Figure 5.4(a).

The test was set up in the following manner. Five different label sets were used with $K = \{2, 3, 5, 10, 20\}$. The initial pixel labels were randomly assigned using a uniform distribution. The region size N was taken from the set {16, 64, 256, 1024, 4096}. To ensure that convergence will occur for large regions, one must use increasingly slower schedules (2.24) for decreasing T until $T \approx 0$ and make T larger to allow for a longer time at higher temperatures (in order to search a larger portion of the solution space Ω_π). Different schedules were tried with $T \in \{1, 3, 10, 30, 100, 300, 1000, 3000\}$ and $\kappa \in \{0.9, 0.93, 0.99, 0.993, 0.999, 0.9993, 0.9999, 0.99993, 0.99999\}$. This results in 1800 tests which were carried out in order of least computationally expensive to most.

The test was carried out in order to determine the minimum computational complexity to obtain global convergence (*i.e.*, the label field populated with only one label). 40

	Number of Labels				
Region Size	2	3	5	10	20
4x4	$2.19 \cdot 10^2$	$2.59 \cdot 10^2$	$4.55 \cdot 10^2$	$3.06 \cdot 10^2$	$4.03 \cdot 10^2$
8x8	$1.43 \cdot 10^3$	$5.23 \cdot 10^3$	$6.10 \cdot 10^3$	$7.22 \cdot 10^3$	$1.06 \cdot 10^4$
16x16	$6.88 \cdot 10^4$	$7.42 \cdot 10^4$	$2.19 \cdot 10^5$	$2.51 \cdot 10^5$	$2.78 \cdot 10^5$
32x32	$1.11 \cdot 10^6$	$1.25 \cdot 10^6$	$8.15 \cdot 10^6$	$1.02 \cdot 10^7$	$1.12 \cdot 10^7$
64x64	$4.87 \cdot 10^7$	$5.41 \cdot 10^7$	$5.82 \cdot 10^7$	-	-

Table 5.1: Number of site visits for convergence times for 2-D walk of annealing. For comparison, on a SPARC-10 running at 333 MHz approximately $25,000$ site visits were carried out per second on a 10-label problem taking approximately $2,400$ seconds.

trials were run for each experiment which included varying the region size, the number of labels, and schedule parameters. The schedule parameters were arranged in the order of hypothesized computational time requirement by starting with low T and low κ and progressing to higher values when convergence was not achieved in 35 out of the 40 trials (*i.e.*, zero energy was achieved most of the time). The results in Table 5.1 are given on the total number of site visits.

It is obvious that the computational cost of using the Gibbs sampler to fill ever larger image regions increases dramatically as ever slower annealing schedules are needed for the algorithm to converge successfully. Unfortunately, a direct comparison with the 1-D case is difficult due to the occurrence of deadlocks in 2-D. For higher numbers of labels the complexity is higher since the label space Ω_π is increasing considerably and the algorithm needs to explore a higher number of possible solutions before converging. This results in a very high computational complexity especially for images which contain typically $1,000,000$ pixels or more.

5.2 Nested Aggregation Framework

In this thesis, the computational complexity of Gibbs sampling is decreased through a hierarchy of graph partitions called stochastic nested aggregation. The nested aggregation

framework defines a hierarchy of ever increasing node groupings. The motivation behind this framework is rooted in the high computational complexity of single site discrete state estimation. Images which are characterized by the appearance of large constant or homogenous patches are ideally suited to be segmented using this hierarchical approach. As Table 5.1 indicates, for increasing region size, Gibbs sampling takes an increasingly large amount of time to converge: would it not be reasonable to achieve this convergence one smaller step at a time?

There are several questions that need to be answered before examining the nested aggregation formulation of the Ising/Potts model. First, what is the benefit of a hierarchical method? Next, what is the nature of local minima in the energy function? Consequently, when would either *deterministic* nested aggregation or *stochastic* nested aggregation be preferable? Fourth, are the partitions of each graph in the hierarchy equivalent to each other? Next, is there a stopping criterion? Finally, is the number of colors or labels K important? Should it be fixed or variable?

5.2.1 Hierarchical vs. Flat Field

What is the benefit of a hierarchical method?

For example, consider that an image region with size 64×64 or 4096 pixels needs to be segmented using 5 labels. It would take on average $5.82 \cdot 10^7$ site visits to fill the region with a single label according to the results in Table 5.1. However, if instead of trying to fill in the entire 4096-pixel region, the algorithm tried to fill in only 16-pixel regions, the computation would take a tiny $4.55 \cdot 10^2$ site visits. At that point, there would be only 256 regions in the resulting segmented image and not the original 4096. The same algorithm could be applied again to the segmentation of the 256-*region* graph by considering the regions as single sites. This would result in 16 larger regions which would be grouped using the same algorithm to form a single constant-label region. This would result in an approximate computational requirement of $1.24 \cdot 10^5$ site visits which is a considerable improvement over $5.82 \cdot 10^7$ site visits, effectively reducing the computation from many hours to a few minutes depending on image size and computer speed. This is only possible once we treat pixels and regions as single-site graph nodes at each level.

Although new graphs are formed at each level of the hierarchy, each of those graphs is

related to the initial or finest level graph in that each node in a higher level graph represents one or more nodes in the lower level graphs. Therefore, adjacent graph nodes or vertices that should be identically colored or labelled are gradually aggregated without merging the vertices that should remain separated according to the model used. Because the connected subgraphs at a level of the hierarchy are produced as arbitrary concatenations of connected components at finer levels, the resulting connected components can naturally fit those of the data (*e.g.*, image) being analyzed, rather than the poor fit of predefined square regions in a coarse-to-fine regular hierarchy. That is, irrespective of the their shapes, the connected nodes at lower levels in the hierarchy fit exactly within the connected subgraphs of at higher levels. Furthermore, given the reduction in the size of the graph at each subsequent level, the hierarchical framework allows for ever faster optimizations at successively higher levels in the hierarchy thus reducing the computational complexity to a fraction of the running time for the single site Gibbs sampler.

Let $s \geq 0$ be the level indicator within the hierarchy where $s = 0$ corresponds to the finest or first level (*i.e.*, for an image segmentation problem this is usually the original image or some preprocessed image patches). Let $\mathcal{G}^{(s)}$ define the graph at level s with $\mathcal{G}^{(0)}$ corresponding to the original graph being partitioned [99]. Each level of the hierarchy is now reformulated as a new graph $\mathcal{G}^{(s+1)}$ based on node aggregation in graph $\mathcal{G}^{(s)}$. Let $\mathcal{V}^{(s)}$ be the set of all partitions at level s such that $|\mathcal{V}^{(s)}| \geq |\mathcal{V}^{(s+1)}|$, *i.e.*, each higher level has fewer possible partitions than the previous one since there are fewer nodes at each successive level. Once $|\mathcal{V}^{(s)}| = |\mathcal{V}^{(s+1)}|$, the process stops.

This mechanism allows for faster processing of the original nodes in $\mathcal{G}^{(s)}$ since $\mathcal{G}^{(s+1)}$ contains fewer nodes each of which represents a subset of the nodes in $\mathcal{G}^{(s)}$. This process is related to graph contraction or decimation [12, 99]; however, instead of carrying forward a subset of the same nodes [99], the new reduced graph $\mathcal{G}^{(s+1)}$ created from the original graph $\mathcal{G}^{(s)}$ will contain some new nodes (merged regions) and edges that are deduced from the nodes and edges of the previous graph and some of the same nodes (regions that did not want to merge) and edges.

An $n^{(s)}$-partition of $\mathcal{G}^{(s)}$, where $n^{(s)}$ is the number of partitions at each level s, is denoted

by

$$\begin{aligned} \pi_n^{(s)} &= (V_1^{(s)}, V_2^{(s)}, \ldots, V_{n^{(s)}}^{(s)}), \\ \cup_{i=1}^{n^{(s)}} V_i^{(s)} &= \mathcal{V}^{(s)}, \\ V_i^{(s)} \cap V_j^{(s)} &= \emptyset, \forall i \neq j. \end{aligned} \tag{5.3}$$

where $n^{(s)} = |\mathcal{V}^{(s)}|$, $n^{(s)} > n^{(s+1)}$ and $n^{(s)} \leq N$. The label field at each level s is described by $\ell^{(s)}$.

Hierarchies allow us to improve both deterministic and stochastic discrete-state estimation techniques in significant (but different) ways. For deterministic discrete-state estimation (*e.g.*, iterated conditional modes), hierarchies of estimates eliminate most problems due to local minima by breaking the deadlocks as shown in Figure 5.4. By using a reduced graph $\mathcal{G}^{(s+1)}$ to represent the original graph $\mathcal{G}^{(s)}$, local minima created through labelling deadlocks are avoided since $\mathcal{G}^{(s+1)}$ no longer has the structure of $\mathcal{G}^{(s)}$. In the case of Figure 5.4(a), instead of the 12 nodes in $\mathcal{G}^{(s)}$ half labelled A and the other half labelled B, the new graph $\mathcal{G}^{(s+1)}$ would only have *two* nodes labelled A and B. The next step would be to either merge the two nodes or leave them depending on the underlying data (pixels) characteristics and the model being used for graph partition.

The labelling deadlocks are caused primarily by the model being used. In this case the Ising/Potts model of the first order computes the energy based on the 4-pixel neighborhood (assuming a regular lattice $\mathcal{S}$). If all edge weights between neighboring pixels are equal (*i.e.*, a region of constant intensity or feature) and if three of the four neighboring labels are identical, then it is easy to see that a vertical or horizontal edge could be created as in Figure 5.4(a) since the central pixel wants to keep its existing label. In the case of real world images, the first reduced graph and all the ones afterwards $\mathcal{G}^{(s)}$, where $s \geq 1$, will be created on an irregular lattices where other graph structure-dependent deadlock formations are possible. For example, Figure 5.4(b) illustrates a deadlock for some pixels with three neighbors. Those deadlocks will be broken again when adjacent similar nodes are merged into a new reduced size graph. Therefore, hierarchies allow Gibbs sampling to converge closer to the globally optimal joint probability $P(\ell)$ although there is no guarantee that the global optimum will be reached especially with deterministic algorithms such as ICM and stochastic ones such as SA when an exponential temperature schedule is used (as will

always be the case in this thesis).

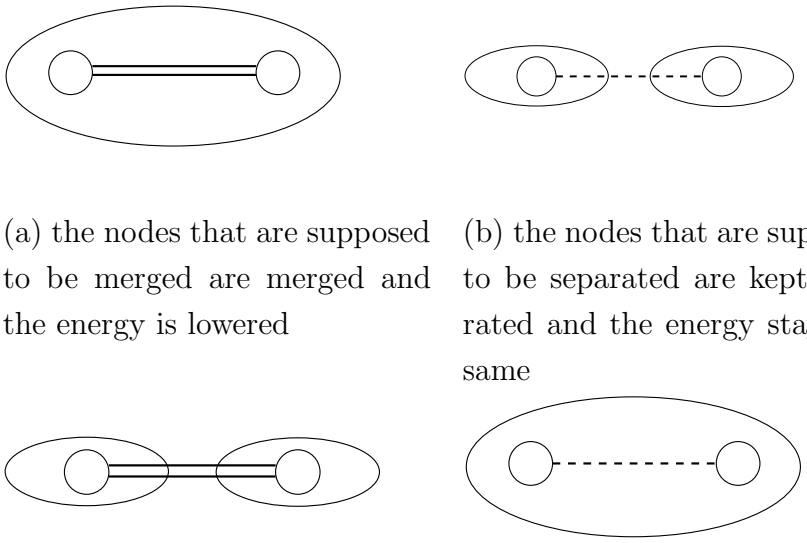
Let us delve deeper into the workings of the framework. Consider that a *level* of the hierarchy is finished after a number of iterations of an algorithm such as ICM or SA. Once the algorithm has converged at a level, adjacent nodes $v_i^{(s)}$ with the same label which form partition $V_j^{(s)}$ are gathered and turned into a new node $v_j^{(s+1)}$ in the reduced order graph that follows; *i.e.*, $V_j^{(s)} = \cup_i v_i^{(s)}$ for all nodes i which are part of the partition j since $V_j^{(s)} \rightarrow v_j^{(s+1)}$. Thus, a new level in the hierarchy is generated and the next pass through the data will use the labels deduced from the previous level to obtain a new set of estimates. The labels obtained at the end of any level are used in subsequent levels with no possibility of reversing results (an approach where reversible moves are allowed at all levels might be possible but is beyond the scope of this thesis); *i.e.*, the framework is not ergodic once the transition from $\mathcal{G}^{(s)}$ to $\mathcal{G}^{(s+1)}$ occurs. However, it is ergodic and moves are reversible (only is using SA) within any level s by definition.

5.2.2 Nature of Local Minima

Is it possible to reach an inappropriate local minimum on any particular level in the hierarchy?

If yes, what steps can be taken to minimize this? The local minima in the Potts model depend on each image being analyzed. There are four possible label configurations between two nodes which can be reached at any time. Figure 5.5 contains an illustration of all these cases. When segmentation proceeds in a desirable fashion cases (a) and (b) occur. In an non-hierarchical algorithm, case (c) leads to a local minimum which is called over-segmentation in image processing. It might be possible to rectify this error in a post-processing step. Of course, in a hierarchical algorithm such as nested aggregation, this merging omission is remediated at one of the subsequent levels. Case (d), however, is the least desirable as it automatically leads to an erroneous partition of the graph which cannot be remediated at in the next levels of the hierarchy. The algorithm should be prevented from reaching such "illegal" configurations. In image processing terms, this is an under-segmentation result which is not a desirable outcome.

Since the results are fixed after convergence is achieved at each level, under-segmentation cannot be reversed. Under-segmentation could occur for several reasons:



(a) the nodes that are supposed to be merged are merged and the energy is lowered

(b) the nodes that are supposed to be separated are kept separated and the energy stays the same

(c) the nodes that are supposed to be merged are not merged and the energy stays the same (over-segmentation)

(d) the nodes that are supposed to be separated are merged and the energy is increased (under-segmentation)

Figure 5.5: There are four possible outcomes when considering two adjacent nodes. Consider diagrams with two adjacent nodes linked by a double line indicating that they should be merged and a dotted line indicating that they should be kept separate. The circles around individual nodes indicate that the algorithm will separate the nodes while an ellipse around both nodes shows that the nodes will be merged by the algorithm. Outcomes (a) and (b) are desirable in that they lead us closer to a local minimum. Option (c) keeps the energy at the same level and delays a possible merging of the nodes to a higher level in the hierarchy. Option (d) is an illegal configuration which leads to a wrong solution as it increases the energy and cannot be corrected at coarser scales.

a label increasing the energy is used due to simulated annealing, or the critical slowing down phenomenon due to labelling conflicts with a limited number of labels is encountered forcing a higher energy state even for deterministic algorithms, or region-to-region spilling (inherently due to model formulation) occurs which is undesirable. For solutions related to the critical slowing down phenomenon due to the number of labels see the detailed discussion in Section 5.2.6 below. In the case of simulated annealing, it is necessary to run the algorithm for several iterations at $T = 0$ in order to help prevent the choosing of an illegal configuration as shown in Figure 5.5(d). Effectively the iterations at $T = 0$ force the labelling configuration into a local minimum (unless the number of labels is too small). To ensure a smooth run for SA from $T > 0$ to $T = 0$, schedules (2.24) (see page 27) are chosen so that $T_{\text{last}} \approx 0$.

It is difficult to devise an appropriate model for segmentation which might avoid region-to-region spilling (see Section 3.3 for a detailed discussion) especially if spilling corresponds to an optimal result for that model. This model would need to have an appropriate form (such as the trade-off between node similarity and a smoothness factor in the Potts model) with an appropriate distance measure for the problem. Distance measures are the focus of considerable research (see Chapters 4 and 6). In the event that the model does not perfectly fit the problem (which is often the case), but still reflects the general correct solution space for the problem at hand (such as piecewise constant Potts/Ising models for image segmentation), enhanced optimization strategies might be used. Some of those possible strategies will be detailed in Section 5.3.

5.2.3 Stochastic vs. Deterministic Optimization

When would either *deterministic* nested aggregation or *stochastic*[4] nested aggregation be preferable?

To answer this question, it is necessary first to determine whether the global optimum of a non-convex energy function can be easily found. If the image contains well-defined edges that separate all homogenous regions, then a deterministic (or "greedy") method will be successful with a good initial labelling $\ell^{(0)}$. For images with well-defined large flat

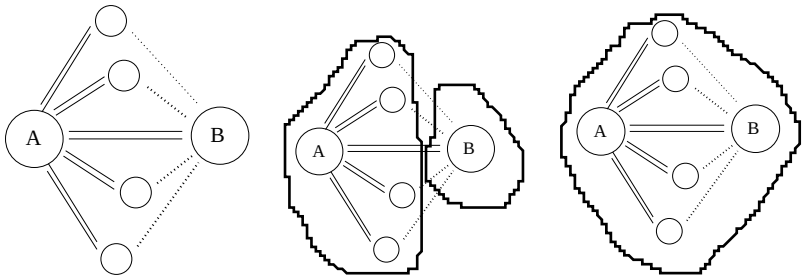
[4]See definitions of deterministic and stochastic on page 24.

areas (*e.g.*, large homogenous regions), nested aggregation can break label deadlocks within those areas and bring about a solution irrespective of the initial labelling $\ell^{(0)}$.

However, in the vast majority of images edges are not well-defined and post-processing algorithms for edge detection such as edge linking [4] need to be applied to create a close-to-ideal energy function where a global optimum is found in an easier fashion than in the original problem. In those cases, a stochastic algorithm will be preferred since it is capable of escaping local minima. Edge linking and other similar algorithms are prone to errors as they usually operate without any prior knowledge of which region edges should be linked and which should not (this is a chicken and egg problem). We will not be using these edge detection postprocessing algorithms as they are irreversible and introduce errors in the segmentation. If we have imperfect edges, we will assume the function we are optimizing is non-convex.

Stochastic algorithms have often been shown to be better better than deterministic ones at solving non-convex problems at a considerable computational cost. This is easily shown to be true by a counter example. Assume that regions in an image are connected through the graph shown in Figure 5.6(a). The solution which would minimize the energy would consist in merging node A with the smaller regions and keeping it separate from node B as shown in Figure 5.6(b). Another possible solution would result in an increase in energy through the merging of all nodes into one as illustrated by Figure 5.6(c). The solution minimizing the energy is only achievable through a stochastic search algorithm since to merge A with the small regions and *not* with B would require a temporary increase in the energy only achieved via some kind of *stochastic* gradient descent while merging A and B results in an immediate drop in energy. This situation is typical of image segmentation problems where a few edge pixels connect two distinct regions, for example see Figure 5.7).

However, in a nested aggregation framework the differences between deterministic and stochastic optimization are not as significant as when using only a single scale. Consequently, we will use both SA and ICM in our experiments. SA will be used with a very fast cooling schedule for T. The idea is to allow the method to increase energy some of the time in order to find a potentially better solution. In algorithms where edge linking is used to avoid region spilling, a deterministic method might be preferable due to its speed given that ambiguous edges have been eliminated. ICM will be used for comparison purposes



(a) (b) (c)

Figure 5.6: A stochastic algorithm will be better than a deterministic one for optimization since it allows us to escape local minima. (a) Consider the above graph where double lines represent nodes that want to merge ($\Phi = 1$) and dotted lines represent nodes that do not want to merge ($\Phi = 3$) and an edge penalty $\beta = 2$. (b) Then, the optimal partition of the graph applying the Potts model would have node A merge with the small nodes and nodes B be separated from them. (c) A suboptimal result would have all nodes merging into one region.

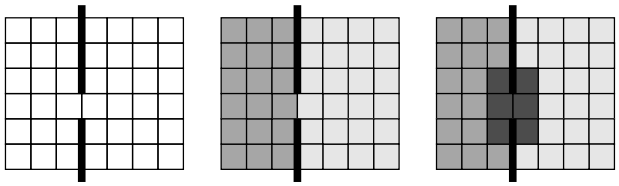
since it gives results at a fraction of the computational cost. Furthermore, the ability to globally converge due to breaking deadlocks is highly desirable in a deterministic algorithm and will be investigated thoroughly.

5.2.4 Equivalence of Graph Partitions

Are the partitions of each graph $\mathcal{G}^{(s)}$ for $\forall s \geq 0$ equivalent to each other?

In other words, does each of the reduced order graphs $\mathcal{G}^{(s+1)}$ have the same global minimum and therefore the same optimal solution as their parent graphs $\mathcal{G}^{(s)}$? It can be easily shown that the global minimum obtained at the coarsest partition is at least a local minimum of the finest level partition. However, the converse is not true since $\mathcal{G}^{(s+1)}$ is a reduced order graph of $\mathcal{G}^{(s)}$ with fewer minima than its parent graph.

The stochastic nested aggregation framework is akin to a smoothing or regularization process that becomes stronger with every graph $\mathcal{G}^{(s)}$ in the hierarchy. As opposed to *initially* maximally smoothing the energy function in order to create a convex function in Granulated Non-Convexity (GNC) [9], SNA tries to transform a non-convex energy into



(a) (b) (c)

Figure 5.7: Ambiguous region merging due to a weak edge: (a) weak edge in an image between two well-defined regions; (b) pixel across the edge possibly becoming two well-defined regions; (c) pixels across the edge merging into a sub-region leading to region spilling and the merging of all pixels into one region.

a convex one by effectively smoothing the function locally at every level of the hierarchy until it is convex at the highest level. At the end of the convergence process, every node in the final graph is labelled differently and does not want to change its label. The local smoothing of the energy at every level allows the optimization algorithm to escape the local minimum and continue with either deterministic or stochastic gradient descent.

Therefore, a necessary condition for $U^{(s)}(\ell^{(s)})$ and $U^{(s+1)}(\ell^{(s+1)})$ to have the same global minimum would be for the energy described by the nodes $\mathcal{V}^{(s+1)}$ and edges $\mathcal{E}^{(s+1)}$ to be a smoothed version of the energy described by nodes $\mathcal{V}^{(s)}$ and edges $\mathcal{E}^{(s)}$. This "smoothing" is obtained here through between-level transition equations that ensure this equivalency. Since equations are specific to the model being used they will be derived for the Potts model in Section 5.3. These equations should in principle transfer the information encoded in $\mathcal{G}^{(s)}$ without any loss to $\mathcal{G}^{(s+1)}$ thus ensuring that their global minima are equivalent. Therefore, the solution to the final level in the hierarchy will also be the optimal solution for the first level in the hierarchy.

The notion of information *loss* alluded to earlier is an important one. Consider for example, the image shown in Figure 5.7(a) with two regions separated by a strong edge with one point at which this edge becomes weak. If region patches form as presented in Figure 5.7(b) then two regions will form at the next level. However, if the region patches form as shown in Figure 5.7(c), then the whole image will become one region at the next level. From the point of view of the Ising/Potts model, Figure 5.7(c) might present the

optimal solution given that a high β would merge both regions through the gap and a small β might create a multitude of small regions within the larger two regions. In practice, Figure 5.7(b) would be more desirable, but a single Potts model might not exist to actually produce this solution.

To achieve this solution one of two things needs to happen: either edge linking needs to be used to fill in *gaps* in the continuity of the edge or region merging parameters have to be chosen in such a way as to encourage the formation of these weak edges. However, there is no guarantee that all such edge gaps will be filled in or that the linking will be done in appropriate locations [4]. If edges appear that do not exist, a local minimum (and perhaps not a very good one) will be reached. The deficiency lies in the formulation of the model. In the case of [4], the edge linking performed on the Canny edge detection results is critical to the segmentation result.

Furthermore, the second solution seems attractive as it can be integrated into the overall principled framework developed here. This solution would involve creating a model-based parameter schedule as mentioned above and as described later in this chapter. This solution would ensure that region-to-region spilling would not have a significant affect on segmentation results.

Finally, since SNA needs to speed-up the convergence of SA at each level by accelerating the temperature schedule, it is very likely that our model will not necessarily converge at each level to a desirable local minimum and, therefore, ultimately to the energy's global minimum. Therefore, $\mathcal{G}^{(s)}$ for $\forall s \geq 0$ might not have equivalent solutions. For example, if the labelling of $\mathcal{G}^{(s)}$ lends itself to a solution as in Figure 5.7(c), the global minimum at higher levels of the hierarchy will not correspond to the global minimum of $\mathcal{G}^{(0)}$. In other words, the solution based on each intermediate graph $\mathcal{G}^{(s)}$ will depend on its structure. We must accept this limitation in order to obtain practical results.

5.2.5 Stopping Criterion

Is there a stopping criterion?

In some deterministic approaches where the model is not well-defined, such as in watershed-based algorithms [2, 13] or other irregular pyramid schemes [99], a stopping criterion is needed to make sure that all nodes are not ultimately merged into one over-

all region. Here we study MCMC methods that stop when the optimization algorithm has reached the problem's stationary probability distribution function. In other words, an implicit stopping criterion is encoded in the energy model and is not some ad-hoc threshold.

Therefore, there are no explicit stopping criteria needed as problems are formulated using an energy model with a well-defined but unknown optimum point. In SNA, the solution obtained maximizes the joint probability $P(\ell)$ irrespective of the algorithm being used to find this optimum point and in practice to find some point close to to this optimum. Once the algorithm converges to a given labelling or graph partition and partitions do not change between two levels, SNA is said to have converged.

5.2.6 Number of Labels

Is the number of colors or labels K important? Should it be fixed or variable?

In general, it is well established via the *Four Color Theorem* [116] that the minimum number of colors to label a planar graph needs to be at least *four*. Consider the graph in Figure 5.8 where two nodes A and B with differing labels need to be merged while being surrounded by a multitude of neighboring nodes. Changing the labels of A and B so that they are labelled identically and all other nodes surrounding them are labelled differently engenders a critical slowing down of the convergence process which can force the configuration to converge to an undesirable local minimum due to inappropriately assigned labels. Essentially, the mislabelling forces an inadvertent region-to-region spilling effect. This is a problem for both stochastic and deterministic approaches.

Suppose regions A and B have labels l_A and l_B respectively. The M other regions surrounding them have random labels that could include l_A and l_B. Regions A and B are similar in features to each other, $\Phi(A, B) \approx 0$ while being different from the surrounding M regions $\Phi(A, C_j) \gg 0$ and $\Phi(B, C_j) \gg 0 \ \forall j = 1, \ldots, M$. The necessary condition for A and B to merge is that none of the small regions adjacent to A can have label l_B and similarly none of the regions adjacent to B can have label l_A. It is critical that none of the regions C_j adjacent to region A (or B) have label l_B (or l_A) in order to allow region A (or B) to assume label l_B (or l_A) at the next iteration. However, the probability of all labels around A not having label l_B is low (and as the number of adjacent regions grows, this probability decreases) since l_B is a legitimate choice for any of the regions C_j.

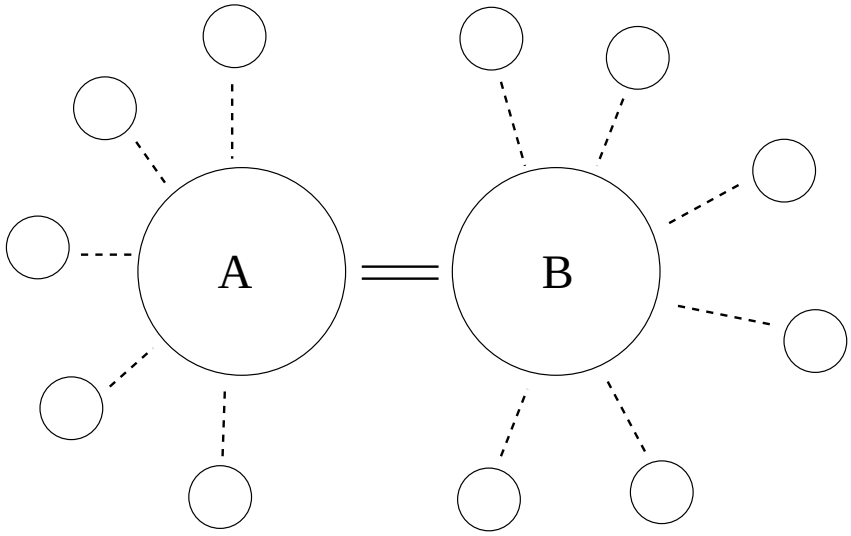


Figure 5.8: Critical slowing down can occur due to a fixed number of labels. If two nodes A and B which should be merged are surrounded by M other nodes $C_j \forall j = 1, \ldots, M$, it might be difficult to assign the appropriate labels to these two nodes without causing one of the nodes C_j to merge into either node A or node B.

If we have K labels and M regions surrounding A and B, the total possible number of labellings is K^{M+2}. The permissible labellings that allow A and B any possible label (from K choices for each node) while nodes C_j any other label (from $N-1$ choices for each of the M regions) giving a total of $(K-1)^M \cdot K^2$ possible permutations. However, the labelling configurations that are allowed need to assign one label to A (from K choices), another different label to B (from $K-1$ choices), and different labels to the regions C_j (from $K-2$ choices for each of the M nodes). Therefore, the total possible number of label configurations that is allowed corresponds to $(K-2)^M \cdot K \cdot (K-1)$. We then take the ratio of these two quantities and obtain the probability of a satisfactory labelling ℓ of all the regions:

$$\tau = Pr(\ell) = \frac{(K-2)^M \cdot K \cdot (K-1)}{(K-1)^M \cdot K^2} = \frac{(K-2)^M}{(K-1)^{M-1} \cdot K} \tag{5.4}$$

which leads to the probability of obtaining the satisfactory coloring after t iterations:

$$Pr(t) = (1-\tau)^{t-1} \cdot \tau \tag{5.5}$$

This corresponds to the geometric distribution. The mean of the waiting times is calculated with

$$\begin{aligned} \mu_{M,K} &= \sum_{t=0}^{\infty} t(1-\tau)^{t-1} \cdot \tau \\ &= \frac{1-\tau}{\tau} \end{aligned} \tag{5.6}$$

and a variance of

$$\begin{aligned} \sigma^2_{M,K} &= \left\{ \sum_{t=0}^{\infty} t^2(1-\tau)^{t-1} \cdot \tau \right\} - \mu^2_{M,K} \\ &= \frac{1-\tau}{\tau^2} \end{aligned} \tag{5.7}$$

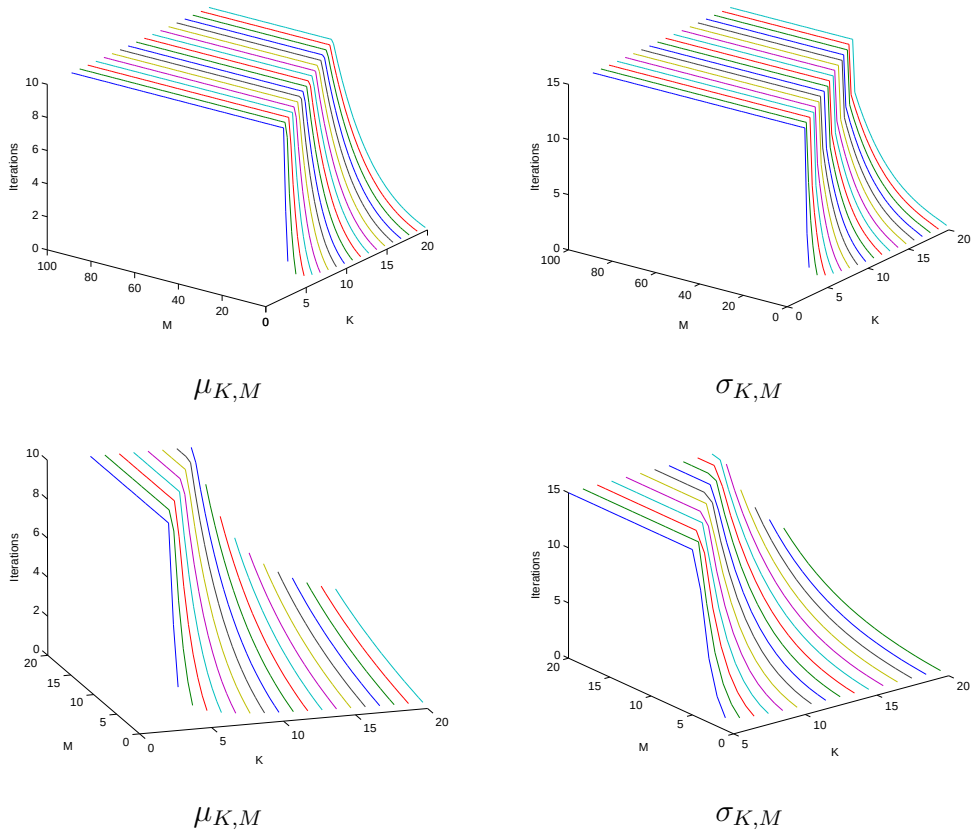
for each $\{M, K\}$ pair. Figure 5.9 shows plots of μ and σ^2 respectively with respect to the number of labels K and number of surrounding regions M. These figures show that as M increases, convergence to a desired label configuration becomes ever more elusive. In other words, the mean number of iterations and the corresponding variance increase drastically

as $M >> K$. This is not surprising given that as $M \to \infty$, $\tau \to 0$ (this decrease can be slowed down with increasing K but not avoided). This result would suggest that it is necessary to keep a high number of labels to avoid falling in an undesirable local minimum or wait a lot of time to force the regions to merge. This is a worst case result as it does not take into account the interactions between pixels due to the Potts model.

There are several solutions to the critical slowing down problem:

- Increasing the number of labels K to a high enough number so that critical slowing down does not occur. This could involve one of a number of strategies. For example, one could experimentally determine the number of labels to match the highest number of edges of a node in a graph of the hierarchy (graphs at higher levels could have nodes that are connected to large numbers of other nodes; *e.g.*, large background areas). By keeping K fixed, the number of labels that are tested for each node is then limited to that number. However, the computational cost could increase dramatically as K is set to a high value. To avoid the increased computational cost of a high number of labels, energy could be computed only for unique labels found in the neighborhood $\mathcal{N}_i$ of node i. At the finest level the highest number of neighbors for a first order neighborhood would always be five (one label for each of the four neighbors and a fifth extra *uncommitted* label in order to allow the nodes to be all disjointed). The number of neighbors would usually increase for higher levels in the hierarchy; however, since the number of nodes decreases, the computational complexity would remain low (usually only a few nodes would need the high number of labels to compute the energy). This strategy allows us to virtually specify an unlimited number of labels. However, one must remember that as the number of uncommitted labels increases the likelihood of selecting a desirable label is lowered (this would always be the case for the finest level and for nodes with few edges at higher levels of the hierarchy). Since this method allows us to increase the number of labels while limiting computation, we will use this strategy in some of our experiments.

- One could also add a label whenever a new label was needed and therefore increase K on a case-by-case basis [4]. Labelling conflicts would be automatically resolved since the new label would, by definition, be different from all other labels. However, every time a new label is added, we would be solving a different problem as the solution



$\mu_{K,M}$ $\sigma_{K,M}$

$\mu_{K,M}$ $\sigma_{K,M}$

Figure 5.9: The mean and variance of the probability distribution function of the number of iterations to converge to a legal label configuration for critical slowing down due to the number of labels. The mean $\mu_{K,M}$ and variance $\sigma_{K,M}$ are plotted as a function of the number of labels $K = \{3, \cdots, 20\}$ and number of small surrounding regions M. The top row shows a result for $M = \{1, \cdots, 100\}$ while the bottom row zooms into results for $M = \{1, \cdots, 20\}$. In the case of $\mu_{K,M}$, the vertical axis indicates the number of iterations necessary on *average* to obtain an acceptable labelling while for $\sigma_{K,M}$ the vertical axis shows the variation about the mean. The functions are capped since there are large differences in τ values. As the number of surrounding regions M increases above the number of labels K, the number of expected iterations to reach an acceptable labelling increases dramatically. This indicates that the number of labels needs to be kept as high as possible.

space would have suddenly increased. Therefore, a long burn-in period would be required in order to obtain a stable large set of labels. Furthermore, this results in a prohibitively expensive Gibbs sampler since the number of computations grows at least linearly with the number of labels. Therefore, we will not be using this method especially because of the long burn-in period required.

- Another way to resolve this critical slowing down problem would be to increase sampling for regions with many neighbors. This would be needed in order to allow those regions to take on other labels after several of its neighbors have changed labels (and not just once in every iteration). This procedure would give the regions with high numbers of connections more chance within an iteration to be labelled appropriately. The number of extra samples drawn from the highly connected nodes could be directly proportional to the number of its edges. However, how many more times should one obtain additional samples? This is an interesting question that is beyond the scope of this thesis and therefore we will not apply this strategy.

- Finally, one could envision an algorithm to make sure that the labels of large and highly connected regions are *non-conflicting*. This could be accomplished with a node relabelling algorithm illustrated in Algorithm 6. Node relabelling would allow the nodes A, B and C_j to acquire a (most likely reduced) set of non-conflicting labels that would then allow some variation in the label assignment at any given level. Node relabelling would proceed according to region size or region connectivity in order to allow the larger and highly connected regions to have non-conflicting labels first. This should then allow Gibbs sampling to avoid the critical slowing down due to conflicting labels. If the relabelling results in a reduced set of labels compared to the original set, then for several iterations after the relabelling, the sampler would have a number of "new" labels to choose from. We will use this method to mitigate the effects of critical slowing down due to the number of labels. Together with a strategy to limit the number of computations to only neighboring node labels, the critical slowing down problem due to the number of labels will be effectively controlled.

  Algorithm 6 is simple and effective. It produces a relabelling using as few as five labels (a more complex algorithm would be necessary to obtain the theoretical limit of four

labels as discussed in the Four Color Theorem [116]). This does not mean however that the total number of labels needs to be five. On the contrary, a higher number of labels would allow the Gibbs sampler to have more flexibility in the assignment of labels after the relabelling step has been completed. However, $K < 5$ would result in many cases in a relabelling failure. We will use this algorithm in many of our experiments.

Algorithm 6 A Node Relabelling Algorithm

1: Set the maximum number of labels for relabelling to K
2: Sort all regions according to size in descending order
3: **for** All regions **do**
4: Assign a label to the current region that does not conflict with neighboring region labels
5: If an appropriate new label cannot be found then stop
6: **end for**

5.3 Hierarchical Bottom-Up Ising/Potts

As an illustration of the method's strength, nested aggregation is applied to the Ising and Potts family of models using Gibbs sampling [48, 88, 4] (other models such as Mumford-Shah [143] can also be used but are beyond the scope of this thesis). We use Gibbs sampling with both a global optimization method like simulated annealing (SA) [76] and a local optimization scheme such as Iterated Conditional Modes (ICM) [88] (*i.e.*, SA with $T = 0$).

The Potts model is one of the models in the family of piecewise constant models [11] which are suitable for image segmentation problems. The first order Potts model is an ideal candidate for the new hierarchical framework as it is fairly easy to analyze, has pairwise cliques and has been used extensively in image processing [11, 48]. Furthermore, it is easy to adapt this model to other pairwise comparison paradigms at higher levels in the hierarchy (see Section 5.4 for specific extensions).

5.3.1 Hierarchical Model Definition

Let's formulate a GRF for segmentation as follows. Given a distance measure Φ, a trivial GRF for segmentation would penalize pixel differences within regions

$$U(\ell) = \sum_{i,j} \Phi_{i,j}\delta_{l_i,l_j}. \tag{5.8}$$

Model (5.8) is flawed, however, since it can be satisfied perfectly by having the region labels $\{l_i\}$ differ for each pair of adjacent pixels, thus segmenting the image into many regions, each one pixel in size. A prior model is required for labels, penalizing too frequent label changes.

In Section 5.1, we have assumed that the data were identical everywhere which resulted in a zero gradient assigned to graph edges between adjacent graph nodes (in equation (5.1), $\Phi(\cdot) \equiv 0$ for all node pairs). Suppose we are now given a data set with a distance measure Φ between adjacent nodes. We can write the Potts model for graph labelling as follows:

$$U(\ell) = \sum_{(i,j)\in\mathcal{E}} \left[\Phi_{i,j}\delta_{l_i,l_j} + \beta_{i,j}(1-\delta_{l_i,l_j})\right] \tag{5.9}$$

where $\Phi_{i,j}$ is the dissimilarity criterion between the features of nodes v_i and v_j, and $\beta_{i,j}$ controls the relative constraints on the degree of region cohesion and fragmentation. This means that $\Phi_{i,j}$ and $\beta_{i,j}$ define relationships between all nodes in the graph (note that they are identically zero for all non-adjacent nodes). $\beta_{i,j}$ can be node-dependent or a constant β throughout the image. In this thesis $\beta_{i,j} = \beta$ is determined experimentally by testing different values on one or more images.

At the finest level, we assume that $\mathcal{G}$ has a first order neighborhood structure on a regular grid shown in Figure 5.4(a). The Potts model then reduces to

$$\begin{aligned} U(\ell) \quad = \textstyle\sum_{i,j} \quad & \Phi(v_{i,j}, v_{i,j+1})\delta_{l_{i,j},l_{i,j+1}} + \Phi(v_{i,j}, v_{i+1,j})\delta_{l_{i,j},l_{i+1,j}} + \\ & \beta\left[(1-\delta_{l_{i,j},l_{i,j+1}}) + (1-\delta_{l_{i,j},l_{i+1,j}})\right] \end{aligned}$$

The basic interaction between β and Φ is illustrated in Figure 5.10.

Models (5.9) and (5.10) are in many ways region growing-type models [137, 59] where decisions to integrate a node into a subgraph are done with respect to the criterion Φ. The major difference between these models and region growing methods is that the first order

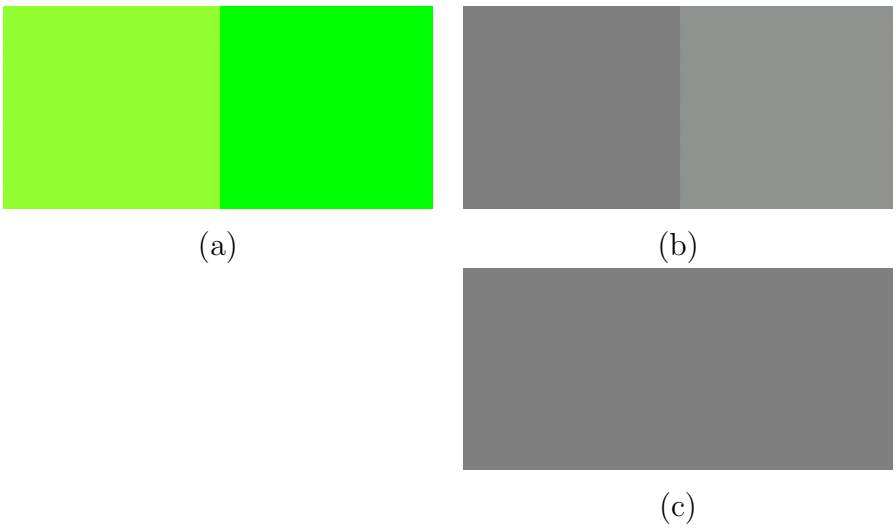


Figure 5.10: Boundary constraints: (a) Reference image; (b) A boundary between two similar green colors is inserted at a cost of $b \cdot \beta$; (c) The boundary between two colors is removed at a cost of $\sum_{(v_{i,j},v_{i,j+1})\in\mathcal{B}} \Phi(v_{i,j}, v_{i,j+1})$ where $\mathcal{B}$ represents the set of b edges. Therefore, if $b \cdot \beta > \sum_{(v_{i,j},v_{i,j+1})\in\mathcal{B}} \Phi(v_{i,j}, v_{i,j+1})$ then the regions are joined together.

Potts model (for a fixed β) is essentially a pairwise energy function in which the inclusion of a pixel or node in a larger region depends only on local comparisons between a *pair* of variables. On the other hand, in region growing algorithms, the inclusion of a pixel or node is non-pairwise as it is very much dependent on the sequence of previously included pixels in that region.

Region-to-region spilling can still occur when using the Potts model. To verify this, the concept presented in Figure 5.10 can be used. Essentially, at any point where $\beta > \Phi$, region-to-region spilling will occur. Regions connected by slowly varying gradients (*i.e.*, regions between which there is no definite or exact edge) or by a small gap tend to be merged (small edge gaps occur due to errors in the camera lens and in the image capture process due to aliasing). This could lead to different solutions unless the model is disambiguated (cf. Section 5.2.2).

Given a distance metric Φ, and a carefully chosen region coupling parameter β, the algorithm performs an over-segmentation of the image by creating a multitude of small, compact regions. In practice, any over-segmentation result like the one in [4] or from a watershed transform [2] can be used as a precursor to the nested aggregation scheme as long as only the desired pixels were grouped (*i.e.*, no regions that straddle borders are present in the initial segmentation). Otherwise, the algorithm will never have the ability to converge to the global minimum (or close to it) since that option would have been preempted in the first processing stage.

To generalize the Potts model to the stochastic nested aggregation formulation, we begin with (5.9), a formulation naturally adapted to an irregular grid [88] to allow an arbitrary number of neighbors, as shown in Figure 5.4(b). The Potts model based on an irregular grid is written as

$$U(\ell) = \sum_{(i,j)\in\mathcal{E}} \{\Phi_{i,j}\delta_{l_i,l_j} + \beta_{i,j}(1-\delta_{l_i,l_j})\} \tag{5.10}$$

where $\beta_{i,j}$ is the region coupling parameter between nodes v_i and v_j. We can easily reformulate (5.10) within a hierarchical fine-to-coarse stochastic nested aggregation framework with

$$U(\ell)^{(s)} = \sum_{i,j\in\mathcal{V}^{(s)},i\neq j} \{\Phi_{i,j}^{(s)}\delta_{l_i,l_j} + \beta_{i,j}^{(s)}(1-\delta_{l_i,l_j})\} \tag{5.11}$$

where $\Phi_{i,j}^{(s)}$ and $\beta_{i,j}^{(s)}$ define node relationships at level s.

Model (5.11) is non-local in that it operates on node aggregates or subgraphs rather than on individual nodes in the original graph $\mathcal{G}^{(0)}$. For boundaries composed of more than one edge at level $s > 0$, $\Phi_{i,j}^{(s)}$ between two regions i and j is an *average* of the individual Φ's; *i.e.*, $\Phi_{i,j}^{(s)}$ is averaged over the length of the boundary. However, (5.11) is still a pairwise model since it compares pairs of regions to each other. A walk of annealing still takes place at the higher levels; however, since there are fewer nodes in the graph, the walk is faster.

5.3.2 Transition Equations Between Levels

In order to validate the hierarchical Potts model (5.11), we need to show that the optimization algorithm applied to it will have the *possibility* to converge to the same result on all levels s of Potts model $U^{(s)}$ (global convergence is only guaranteed if simulated annealing is used with a logarithmic schedule [48]). In other words, $\arg\min U^{(s+1)} = \arg\min U^{(s)}$ needs to be satisfied for $\forall s \geq 0$. This means that the energy function formulations at $U^{(s)}$ and $U^{(s+1)}$ need to be equivalent for the purpose of function minimization.

In Appendix A, we prove that $U^{(s+1)} = U^{(s)} + \bar{U}$ where $\bar{U}$ is a constant representing the sum of all the edges $\mathcal{E}_e^{(s)} = \{(i,j)\}$ to be erased at level s, or $\bar{U} = \sum_{(i,j)\in\mathcal{E}_e^{(s)}} \Phi_{i,j}^{(s)}$ where $\mathcal{E}_e$ is the subset of edges of $\mathcal{E}$ to be erased. This result shows that the minimum of these two functions is essentially the same and therefore their solutions should lead to the same solution. In general, this statement is true; however, the use of a pairwise energy function has some interesting consequences which can prevent a better minimum to be reached. This issue will be discussed shortly; first, transition equations are summarized.

Based on the proof in Appendix A, transition equations from level s to level $s+1$ can be constructed by summing all the $\Phi_{i,j}^{(s)}$ and $\beta_{i,j}^{(s)}$ values that are common between the two adjacent regions. The result will be a new edge in graph $\mathcal{G}^{(s)}$ between the two new regions such that

$$\Phi_{r,r'}^{(s+1)} = \sum_{i\in V_r^{(s)}} \sum_{j\in V_{r'}^{(s)}} \Phi_{i,j}^{(s)} \tag{5.12}$$

and

$$\beta_{r,r'}^{(s+1)} = \sum_{i \in V_r^{(s)}} \sum_{j \in V_{r'}^{(s)}} \beta_{i,j}^{(s)} \tag{5.13}$$

where r and r' are the node indexes in the new graph $\mathcal{G}^{(s+1)}$ while $V_r^{(s)}$ and $V_{r'}^{(s)}$ are the corresponding sets of nodes at level s. We now have model (5.11) which governs how the labelling is done at each level s together with level-to-level transition equations (5.12) and (5.13).

The edge between two new regions at level $s+1$ can easily contain one or more old edges between subsumed nodes from level s. In Figure 5.3(c), the two regions share four node-to-node edges. Once both regions become nodes at the next level (to break the local minimum due to the node deadlock), the relationship between the new nodes should be based on the relationship between the old nodes. Only the edges in $\mathcal{E}^{(s)}$ that survive in $\mathcal{E}^{(s+1)}$ will be retained while all other edges become irrelevant since they have been subsumed inside newly formed regions $V^{(s+1)}$.

To illustrate how the transition equations work consider two levels of the hierarchy shown in Figure 5.11. Consider that nodes I, J, L, and M will be merged into node $IJLM$. When nodes I, J, L, and M are merged into one node, all the relationships between them governed by $\Phi_{r,r'}^{(s)}$ and $\beta_{r,r'}^{(s)}$ must be eliminated since they cease to be individual nodes. The relationships between nodes I, J, L, and M and nodes G and H will now become the relationships between node $IJLM$ and nodes G and H. To accomplish this, we need to sum the appropriate $\Phi_{r,r'}^{(s)}$ and $\beta_{r,r'}^{(s)}$. For example, $\Phi_{G,IJLM}^{(s+1)} = \Phi_{G,I}^{(s)} + \Phi_{G,J}^{(s)}$.

5.3.3 Stochastic Nested Aggregation for the Potts Model

Finally, we can summarize the stochastic nested aggregation algorithm in Algorithm 7. We use simulated annealing with a schedule for temperature T. If one would like to use the deterministic Iterated Conditional Modes algorithm this can be done by setting $T = 0$ for all computations. The relabelling algorithm is optional. Since both algorithms are nested versions of the original SA and ICM, we will refer to them as SNA-SA and SNA-ICM respectively.

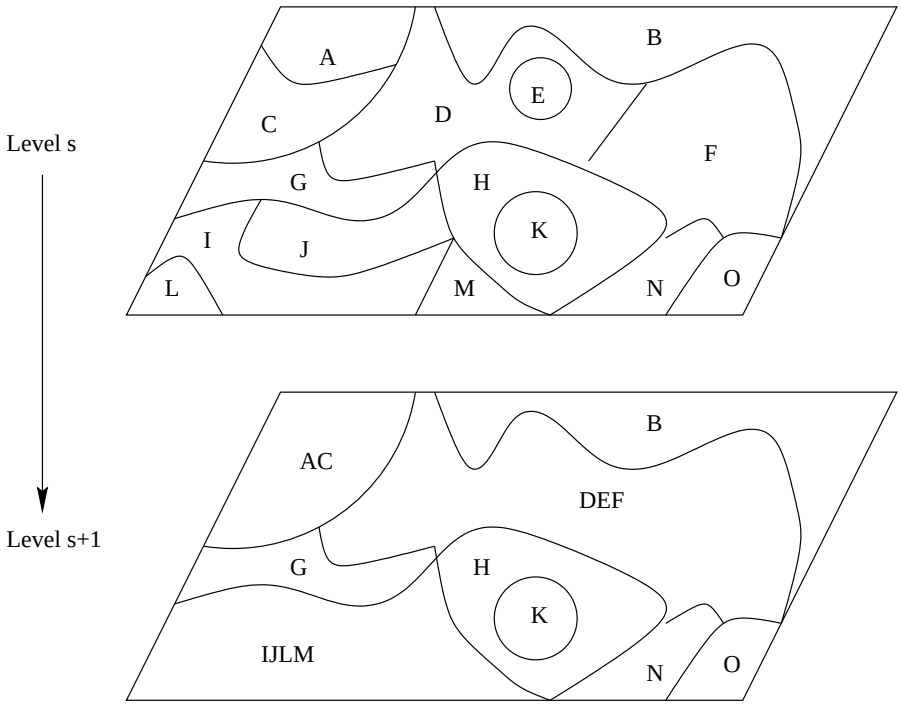


Figure 5.11: Region merging: nodes on Level s are merged into fewer nodes at Level $s+1$.

Algorithm 7 Stochastic Nested Aggregation Graph Partition Algorithm

1: Split the image by assigning random labels to all nodes $\{v_i^{(0)}\}$ or obtaining preprocessed atomic regions $\{V_j^{(0)}\}$
2: **for** $s = 0, \ldots, s_{max}$ (from finest to coarsest) where s_{max} is the dynamically-determined final level in the hierarchy **do**
3: Relabel images according to Algorithm 6.
4: **repeat**
5: **for** $v_i^{(s)}, \forall i$ **do**
6: Minimize the energy in model (5.11)
7: Update the node's label based on Gibbs sampling Algorithm 1
8: **end for**
9: Update T according to the desired schedule (2.24)
10: **until** $|\mathcal{V}^{(s)}| = |\mathcal{V}^{(s+1)}|$
11: Apply transition equations (5.12) and (5.13) to transition from level s to $s+1$
12: **end for**

The algorithm is divided into two parts: a trivial image splitting part in the first step, and a region merging part in subsequent steps. The image splitting part could consist of either considering each individual pixel its own region or by using a preprocessing algorithm (*e.g.*, [4]) in order to create atomic regions.

This algorithm is specified for a Potts model using Simulated Annealing and stochastic nested aggregation. Other stochastic/deterministic optimization algorithms and energy models can be used instead for different types of problems. The same general principles would still apply as long as the problem can be framed as a graph partition problem using pairwise node comparisons.

5.3.4 Computational Complexity

Next, to ascertain the computational complexity of nested aggregation, we consider the Potts model optimized using SNA-ICM and SNA-SA. In Figure 5.2 (see page 77), we have seen initially that simulated annealing applied to the Ising model in 1-D converges approximately in $O(N^3)$ site visits where N is the size of the region (*i.e.*, $O(N^2)$ iterations

per each of the N pixels or nodes).

We can also easily analyze convergence speed for the 1-D hierarchical model. Consider that the effort to group pixels in an image or graph $\mathcal{G}^{(0)}$ of size N into size R regions is $O(N \cdot R^2)$ since we have to group N nodes/pixels each of which will require $O(R^2)$ computations. Then, once we have groups of size R, if we want to group them into groups of size R^2, the effort will be $O(\frac{N}{R}R^2)$ or $O(N \cdot R)$ since we have $\frac{N}{R}$ nodes in graph $\mathcal{G}^{(1)}$. This goes on until we have groups of size $N = R^s$ (the largest possible region in the image/graph is the image itself) where s is the level of the hierarchy. In practice, images seldom consist of a single color or feature and we have $R^s \ll N$

We can summarize the complexity as follows:

$$\left.\begin{array}{lll} \text{Level } 0 & 1 \rightarrow R & O(N \cdot R^2) \\ \text{Level } 1 & R \rightarrow R^2 & O(\frac{N}{R} \cdot R^2) \\ \text{Level } 2 & R^2 \rightarrow R^3 & O(\frac{N}{R^2} \cdot R^2) \\ \vdots & \vdots & \\ \text{Level } s-1 & R^{s-1} \rightarrow N & O(\frac{N}{R^{s-1}} \cdot R^2) \end{array}\right\} O(N \cdot R^2)$$

where the first line represents the work at the finest scale to group regions of size one (*i.e.*, pixels) into regions of size R with an effort of $O(N \cdot R^2)$. The subsequent rows represent the successive grouping of regions into larger regions increasing at each level the region size by a factor of R. Note that level s is the last level of the hierarchy corresponding to the graph partition result. Furthermore, R is not selected by the user. Instead, R is estimated based on the reduction in number of nodes from $\mathcal{G}^{(s)}$ to $\mathcal{G}^{(s+1)}$.

The complexity for the overall process is essentially the same as that for the first level of the hierarchy since the region merging happens in a geometric fashion. Notice that the region computation has a constant factor due to the geometric progression which is summarized by $1 + \frac{1}{R} + \frac{1}{R^2} + \cdots + \frac{1}{R^{s-1}}$ or $\frac{R^{s+1}-1}{R^s(R-1)} \leq 2$ for all $R \geq 2$. For $1 \leq R \leq 2$, $\frac{R^{s+1}-1}{R^s(R-1)} > 2$ will result in a slight increase in the computational complexity. Namely, when $R \approx 1$ for all levels, $\frac{R^{s+1}-1}{R^s(R-1)} \approx s$ which indicates that, the overall complexity will still remain low as long as the number of levels is small with respect to N. For large images (which are of interest to us), this is not a problem as most of the reduction in the number of nodes should happen in the first few levels of the hierarchy which would be followed

by a few levels of small adjustments to region sizes before the algorithm converges to the desired stationary probability.

We contend that R is not a function of N since if we partition the image into smaller pieces, the same geometric reduction of graph size will happen for those images. Furthermore, R grows as a function of the number of site visits at a given level and, when using the Potts model, R grows as a function of β. If β is small, R can approach 1 after very few iterations; therefore, when a long T schedule is used for SA, there are many extra iterations (in comparison to ICM) which do not contribute to increasing R in a significant way resulting in a higher complexity depending on the number of levels s. Note that the number of levels is also not chosen and is a result of the discrete state estimation process.

For ICM, by limiting the number of iterations through the graph to a minimum and, therefore, creating large regions quickly, the complexity of the 1-D nested aggregation algorithm is at most $O(N)$ which is a considerable improvement over $O(N^3)$. In practical terms, the highest computational complexity occurs at the pixel level. Our experiments have been conducted assuming that the model is being applied directly to the smallest possible regions, *i.e.*, pixels (this algorithm could easily be applied to image patches or atomic regions such as in [4]).

When applying nested aggregation to 2-D data like images, the estimation of computational complexity is more difficult. It must be done using a Monte Carlo simulation as it was done for the non-hierarchical annealer in Section 5.1. We performed the following experiment using the stochastic nested aggregation Algorithm 7. As in Section 5.1, five different label sets were used with $K = \{2, 3, 5, 10, 20\}$. The initial pixel labels were randomly assigned using a uniform distribution. The region size N was taken from the set {16, 64, 256, 1024, 4096}. A single value was used for β; however, this value is inconsequential since the regions have $\Phi \equiv 0$.

For the hierarchical test, both SA and ICM experiments were done since now ICM can be used as a global optimization method. For SA experiments, we chose an annealing schedule (2.24) with an initial temperature $T = 0.3$, and an exponential decay of $\kappa = 0.1$. Each level of the hierarchy was run for six iterations (three iterations at $T > 0$ and three iterations at $T = 0$). The iterations at $T = 0$ ensured that the configurations in SA experiments converged to a legal label configuration (effectively a local minimum after

	Number of Labels				
Region Size	2	3	5	10	20
4x4	$2.25 \cdot 10^0$	$2.58 \cdot 10^0$	$4.45 \cdot 10^0$	$2.99 \cdot 10^0$	$3.85 \cdot 10^0$
8x8	$3.61 \cdot 10^0$	$1.29 \cdot 10^1$	$1.50 \cdot 10^1$	$1.72 \cdot 10^1$	$2.53 \cdot 10^1$
16x16	$4.41 \cdot 10^1$	$4.69 \cdot 10^1$	$1.36 \cdot 10^2$	$1.52 \cdot 10^2$	$1.66 \cdot 10^2$
32x32	$1.79 \cdot 10^2$	$1.99 \cdot 10^2$	$1.27 \cdot 10^3$	$1.55 \cdot 10^3$	$1.67 \cdot 10^3$
64x64	$1.97 \cdot 10^3$	$2.16 \cdot 10^3$	$2.28 \cdot 10^3$	-	-

Table 5.2: Ratio of computational complexity between a flat field annealer (*i.e.*, no hierarchies) and a nested annealer (hierarchies via stochastic nested aggregation) using simulated annealing as a function of region size and number of labels. The ratios reflect the computational gain over the results in Table 5.1.

high energy states in SA). For ICM experiments, only $T = 0$ was used and each level of the hierarchy was run for only one iteration (in order to obtain a low R.

The results for the hierarchical annealer reflect convergence in 40 out of 40 trials which means that all experiments converged on the homogenous label field for both ICM and SA. We compute ratios between hierarchical and non-hierarchical convergence times for all image size and number of labels combinations for the SA and ICM shown respectively in Table 5.2 and Table 5.3. Note that now ICM converges to the minimum configuration (in this toy problem) irrespective of initial conditions whereas it was not considered for flat field experiments in Section 5.1 due to getting stuck repeatedly in local minima. For more complex energy functions we will see that there is some dependence on initial conditions however specifying $\ell^{(0)}$ is no longer crucial for ICM's success. This is one of the fundamental results of this thesis.

We can compare results in Table 5.2 and Table 5.3. For an image size of at most 32×32 and using ten labels, SA will be on average 179 times faster than when using a non-hierarchical annealer (shown in Table 5.1). For ICM when using only one iteration at $T = 0$, the same image could be filled in 768 times faster as shown in Table 5.3. This result shows that nested aggregation gives a considerable improvement in speed for both the SA and the ICM algorithms. It is also clear that as the maximum size of a region in an image to be filled in with a single label increases, the benefit of using the hierarchical

	Number of Labels				
Region Size	2	3	5	10	20
4x4	$1.15 \cdot 10^1$	$1.27 \cdot 10^1$	$2.08 \cdot 10^1$	$1.30 \cdot 10^1$	$1.52 \cdot 10^1$
8x8	$2.07 \cdot 10^1$	$7.14 \cdot 10^1$	$7.64 \cdot 10^1$	$8.10 \cdot 10^1$	$1.08 \cdot 10^2$
16x16	$2.59 \cdot 10^2$	$2.65 \cdot 10^2$	$7.23 \cdot 10^2$	$7.39 \cdot 10^2$	$7.36 \cdot 10^2$
32x32	$1.06 \cdot 10^3$	$1.14 \cdot 10^3$	$6.87 \cdot 10^3$	$7.68 \cdot 10^3$	$7.52 \cdot 10^3$
64x64	$1.17 \cdot 10^4$	$1.24 \cdot 10^4$	$1.23 \cdot 10^4$	-	-

Table 5.3: Ratio of computational complexity between a flat field annealer using simulated annealing and nested aggregation using ICM (simulated annealing with $T = 0$) as a function of region size and number of labels. The ratios reflect the computational gain over the results in Table 5.1.

approach augments correspondingly. This is another fundamental result of this thesis.

Note also the significant difference in speed between the SA and ICM results. It is clear that ICM is considerably faster than SA (in this case by roughly a factor of 5). We used a very aggressive schedule for SA to obtain results in Table 5.2. For longer T schedules which might be necessary for complex graphs like images, this difference will be even more pronounced.

5.3.5 Comparison to Existing Acceleration Methods

Stochastic Nested Aggregation shares similarities and differences with other approaches described in Section 3.5.3.

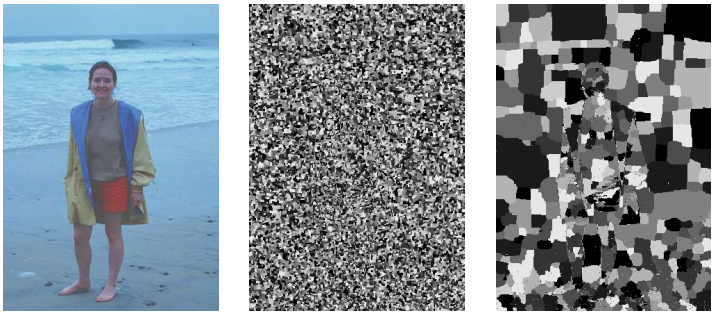
Stochastic Nested Aggregation generalizes previous work in irregular bottom-up approaches such as hierarchical watersheds [2, 13], one-at-a-time region merging [25, 94], irregular pyramids [99, 12] and highest confidence first [25, 94]. First, stochastic nested aggregation works directly with either stochastic or deterministic methods; whereas all methods reviewed in Section 3.5 are deterministic in nature. Second, SNA is a hierarchical method that stops when the stationary joint probability distribution function has been reached whereas all other methods produce an irregular pyramid that ends in a single node

[99, 12]. Under certain conditions, the maximum joint probability of the coarsest level is the same as that of the finest level and thus SNA endeavors to produce hierarchies that end with the "optimal" label configuration that is a solution to the finest level. Third, we introduce a lower bound on the number of labels to be used and show that a critical slowing down due to a low number of labels can occur. All methods either use an infinite number of labels or some experimentally determined number without any justification. Fourth, SNA is applicable to any model in which the various levels of the hierarchy are optimizing the same energy function. Other methods are dependent on a particular data representations and are not easy to generalize to other frameworks or to analyze. Finally, SNA bears some resemblance to Carballo's hierarchical network flow approach [17] which achieves a speed up of at least a factor of 10. Within each regular block there exists an irregular partition of the subgraph with possibly many different nodes which are merged with irregular partitions of the adjacent blocks.

Stochastic Nested Aggregation is similar to the graph cuts and cluster sampling frameworks in that both are irregular grid-based methods. Graph cuts [11, 142] carries out a progressive subdivision of an image or graph based on a unique global criterion from the top down. The global criterion is usually the same one which we use (*i.e.*, the Potts model) [11]; however, the direction of the partitioning is different. The speed-up of the single site annealer is considerable resulting in a partitioning algorithm with $O(N)$ complexity [11] which is comparable to our bottom-up irregular approach.

Cluster sampling partitions the space of piece-wise constant functions such as the Potts model by dynamically splitting, merging and regrouping sizeable subgraphs of the image. As the partitioning progresses the subgraphs become gradually larger. The main advantage of cluster sampling over graph cuts and nested aggregation is that it simulates ergodic and reversible Markov chain jumps in the space of all partitions. Therefore, cluster sampling can reverse a move whereas in hierarchical approaches moves are only reversible within a level of the hierarchy and not between levels.

Stochastic Nested Aggregation is very different from top down regular grid-based techniques [72] which produce undesirable blocky partitions without a considerable speed up.



Original image ICM SA

Figure 5.12: ICM converges to a local minimum. SA also converges to a local minimum with an exponential T schedule. This result is a better than the one arrived at with ICM.

5.3.6 Preliminary Results

We will motivate results by first showing image segmentation carried out using ICM and SA. Figure 5.12 presents an example of using ICM and SA on a Potts model. The label deadlocks are clearly present throughout the image for the ICM result. SA achieved a better result with an long exponential schedule (but obviously not long enough since a desirable local minimum has not been reached!). An optimal solution obtained with SA is not presented here due to extremely high computational complexity. It would have taken approximately more than 2×10^{10} site visits to converge or more than one day of processing[5]!

Figure 5.13 gives an illustration of SNA-SA with $K = 10$ labels[6]. SNA-SA was set up to run with three iterations at a non-zero temperature (for schedule (2.24) starting with $T = 1.0$ and $\tau = 0.9$) and three iterations at $T = 0$. With this schedule, SNA-SA ran usually between one and two minutes due to the long T schedule. One can easily discern which pixels at one level were merged into regions at the next level. Depending on the model and T schedule, there may be more or fewer levels. The fewest levels will be obtained for very small and very large β (small β discourages merging and large β merges all nodes)

[5]All running time results in this section and all subsequent sections are given for C code run on a 2.4 GHz Intel Pentium-based PC.

[6]Note that most segmentation results are usually shown using five labels as they are captured after the relabelling scheme has been applied which reduces the number of labels to four or five.

with the largest number of levels somewhere between the two β extremes. Note that Besag's coding method was implemented [6] for all levels in the hierarchy (cf. Section 2.9). At level $s+1$, a coding corresponded to the graph partition obtained at level s such that all regions given label $k \in \mathcal{L}$ were assigned to coding $S_k^{(s+1)}$. There were as many codings as distinct labels. Note that due to relabelling which produces a graph with only four or five distinct labels, the number of codings remained small.

The assumption of only grouping nodes which should be grouped at each level s is paramount and depends very much on the type of model and the underlying problem. In the case of using the Ising/Potts model for image segmentation, the assumption holds only if at each level of the hierarchy $\beta_{i,j} < \Phi_{i,j}$ for all pixels where placing an edge would lead to the optimal solution.

From this point of view, determining the appropriate β for segmenting each image is important. β and sums of β (when edges between regions are longer than one pixel) will determine what should and should not be grouped at each level. In this thesis, β is chosen experimentally for each image in order to produce a desirable segmentation result. The automatic estimation of β is a topic that is beyond the scope of this thesis.

Consider the results presented in Figure 5.14 and Figure 5.15 showing image segmentation for different β for SNA-ICM and SNA-SA respectively[7]. SNA-ICM was set up to run for two iterations at each level thus providing the fastest possible convergence times (see the computational complexity discussion in Section 5.3.4). SNA-ICM results ran on average in less than a minute (the algorithm ran faster for higher values of β since more nodes were being merged at the lowest level) with no special preprocessing (*e.g.*, edge detection, clustering, etc.).

These figures show clearly that applying stochastic nested aggregation to either ICM or SA for the Potts model leads invariably to region-to-region spilling for higher values of β. Note that different results might be obtained for models other than Potts and therefore we limit this analysis to the Potts model. Qualitatively the spilling is not as severe for SNA-SA as it is in SNA-ICM, especially if one compares the results for each β value in Figures 5.14 and 5.15. At the lowest value of β, more small regions form and

[7]Result images are indicative of general results for the algorithms/models and are not chosen to show a "nice" result.

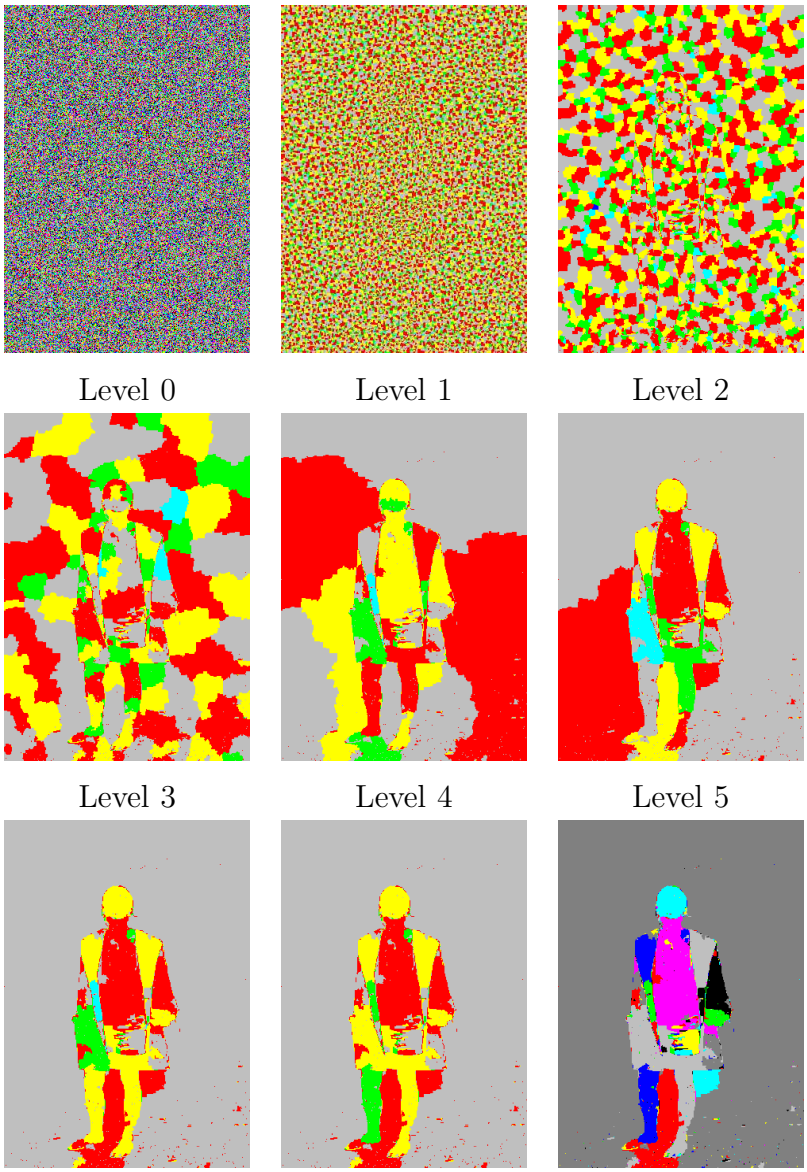


Figure 5.13: Detailed level by level results for SNA-SA using vector angle in model (5.11) with a $\beta = 0.0045$ model.

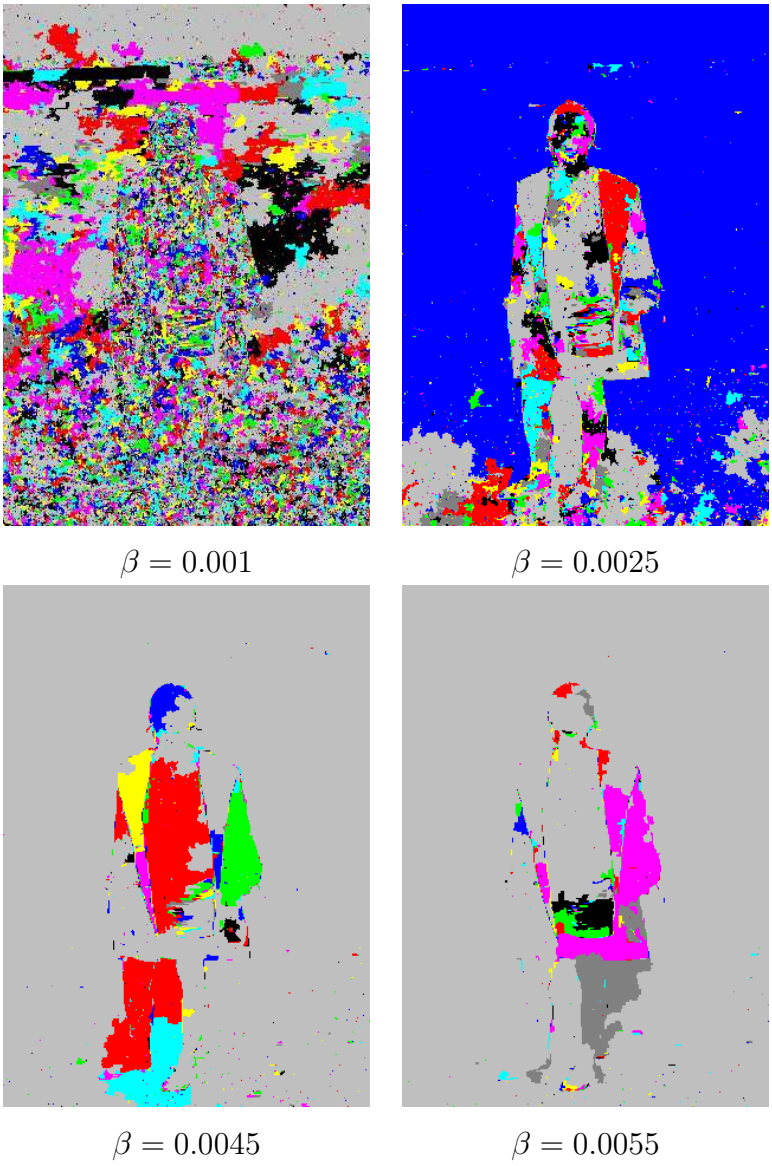
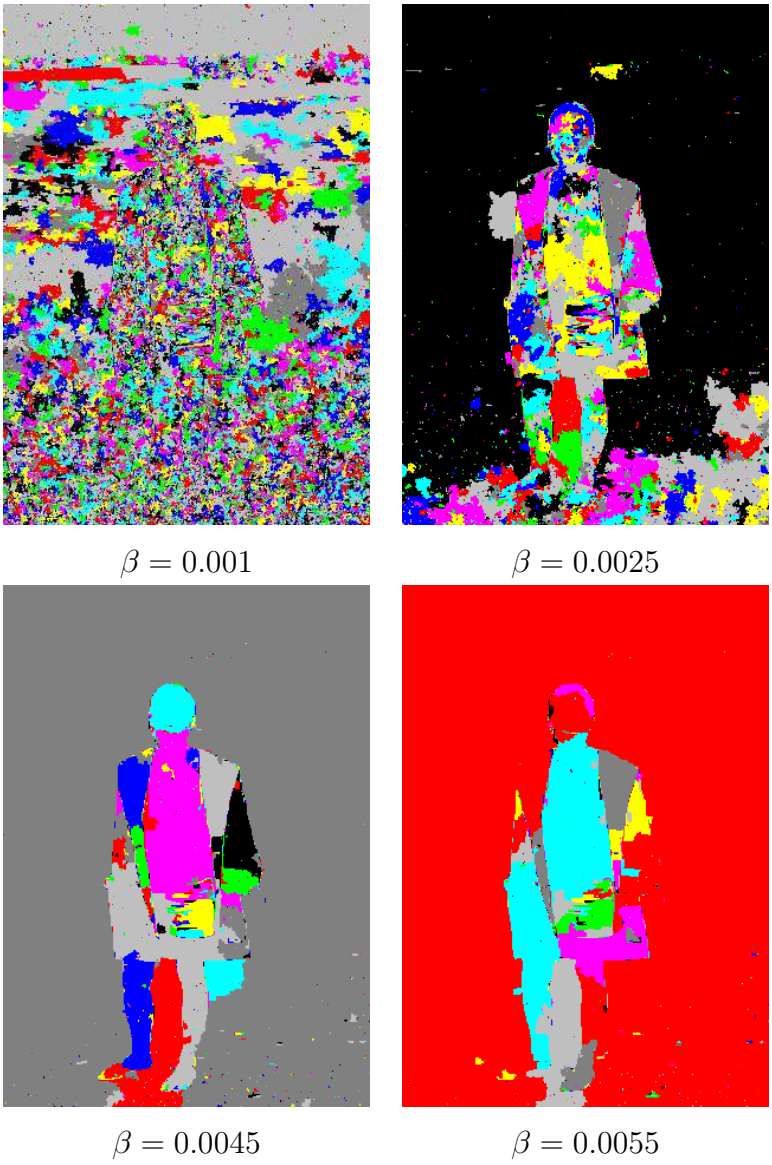


Figure 5.14: SNA-ICM results using vector angle in model (5.11) with different β values. Region spilling occurs for the different models starting with $\beta = 0.0025$ where region spilling occurs (top of the left shoulder) while at the same time major image components (near the bottom of the image) are not merged into one region.



$\beta = 0.001$ $\beta = 0.0025$

$\beta = 0.0045$ $\beta = 0.0055$

Figure 5.15: SNA-SA results using vector angle in model (5.11) with different β values. Region spilling occurs for the different models starting with $\beta = 0.0025$ and especially for $\beta = 0.0045$ (foot at the bottom of the image, parts of the jacket) while at the same time major image components (red area at the bottom of the image) are not merged into one region. However, results appear less prone to region spilling than SNA-ICM.

the image segmentation result looks very fragmented and no spilling occurs. However, as β is increased, the fragmentation slowly disappears and region spilling starts to occur with increasing frequency. At very high values of β, large distinct regions start to merge with each other. Ultimately, choosing a very high value of β results in all regions merging together. Qualitatively, those preliminary results obtained using SNA-SA appear better than those obtained with SNA-ICM. This is most likely due to SA's ability to find lower minima by going to a higher energy state with non-zero probability (see Section 2.8).

In summary, for higher values of β, the nodes in the graph will have a higher likelihood of merging. The spectrum of β values ranges from a very low value (*e.g.*, lower than the smallest $\Phi_{i,j}$ in the graph) when all pixels are individual regions to a very high value (*e.g.*, higher than the largest $\Phi_{i,j}$ in the graph) when all pixels merge into one node or pixel. The desirable β is usually somewhere between those two extremes.

Figure 5.16 shows the end result of SNA-SA runs with different temperature schedules (2.24) all for $\beta = 0.0045$. As temperature schedules become longer, we can say qualitatively that results improve. This is an expected result as longer schedules lead to results closer to the global optimum [48]. However, due to the persistence of region spilling, longer schedules do not seem to warrant the significantly increased computational cost. It should be added that the number of iterations at each level is not important for ICM from the point of view of convergence, as it performs gradient descent; *i.e.*, always choosing a better local optimum point at each iteration. The fewer the iterations at each level, the faster SNA-ICM will be since the algorithm will quickly benefit from the increased region/node size. The only variable will be the structure of $\mathcal{G}^{(s)}$ which will influence to some degree the type of local minimum reached.

5.3.7 Preventing Region-to-Region Spilling: Graduated Models

We now come back to discussing the issue of a pairwise energy function. Consider the artificial image in Figure 5.7 (see page 89) reflecting an ambiguous region-to-region spilling scenario. A small gap exists in the (much longer) edge between the two regions. In other words, $\beta_{i,j} > \Phi_{i,j}$ for edges $\{(i,j)\} \in \mathcal{E}_{\text{gap}}$ where $\mathcal{E}_{\text{gap}}$ represents all edges in the graph that define the gap. Because the energy function is pairwise, the labels are decided by considering the pairwise relationships of one pixel with its neighbors at any one time

$T = 1, \tau = 0.7, iter = 3$

$T = 3, \tau = 0.7, iter = 6$

$T = 15, \tau = 0.7, iter = 10$

$T = 100, \tau = 0.7, iter = 20$

$T = 100, \tau = 0.8, iter = 40$

$T = 100, \tau = 0.9, iter = 70$

Figure 5.16: SNA-SA results using vector angle in model (5.11) with $\beta = 0.0045$ run using different T schedules (2.24). It is clear that region spilling occurs for the different temperature schedules (even for the last one). However, the results indicate that a slower T schedule can improve results at the cost of computational complexity. The SNA-SA using the first schedule of $(T = 1, \tau = 0.7, iter = 3)$ ran on average in approximately 60 seconds while using the last tested schedule of $(T = 100, \tau = 0.9, iter = 70)$ took approximately 490 seconds which was primarily due to the 70 iterations at the finest scale.

without simultaneously considering other pixels (and therefore non-pairwise relationships). This leads to the gap being breached since both pixels on either side of the edge want to merge. These nodes also want to merge with their other neighbors and so on.

However, placing an edge in this small gap would produce a lower energy than carrying out the region-to-region spilling. Inserting an edge in this small gap could only be guaranteed with the current methodology if a non-pairwise function was used since deciding which "gap" to fill-in is a non-local problem. However, this might be difficult as SNA is based on the premise of pairwise comparisons. It might be possible to apply SA with a very slow schedule; however, there is no guarantee that region-to-region spilling will not occur since a result containing region spilling might be the optimum point for SA. However, there are ways to *simulate* non-local interactions in the current framework by using region prototypes (see Section 5.4).

Another way would be to *guide* the simulated annealer to closer to the global minimum via edge linking [4]. However, edge linking fixes region edges and if a wrong decision is made, that decision cannot be undone. We can call this explicit edge linking. However, there is potentially, a better way to carry out the same operation using implicit edge linking. Let's examine again Figure 5.14 and Figure 5.15. Would the edges in the results for model $\beta = 0.001$ contain the edges of the results for model $\beta = 0.0025$ and so on?

Suppose, one applies thresholds $\iota^{(1)} = 0.001$ and $\iota^{(2)} = 0.0025$ to a graph with pixel differences Φ. The edges that will appear for $\iota^{(1)}$ will certainly contain the edges for $\iota^{(2)}$ since all values of $\Phi < \iota^1$ also satisfy the relationship $\Phi < \iota^{(2)}$; however the converse is not true. One could then build a hierarchy of nested thresholds $\iota^{(s)}$ in order to successively eliminate edges from the graphs on lower levels of the hierarchy.

Because Φ is averaged over the length of the boundary at higher levels of the hierarchy, we tend to believe Φ more for large regions than for small ones. Furthermore, inappropriate merging or region spilling occurs more frequently at the lower level of the hierarchy since the edges at those levels are shortest and therefore least certain. Therefore, to minimize the likelihood of merging two regions through a "gap," we need to use a very conservative β at the finest level. At higher levels, β would be progressively relaxed as the boundaries increase in length. Thus, by using a series of nested β's in this fashion, we create a series of nested models.

Nesting models introduces a β schedule for the Potts model such that $\{\beta_0, \ldots, \beta_b, \ldots, \beta\}$ is a set of different β_b values with the characteristic $\beta_b < \beta_{b+1}$ and where β is the desired model. In other words, we start with a small edge penalty and increase it according to the schedule until we reach the desired edge penalty β.

Introducing a β schedule which we also call a set of Graduated Models (GMs) has several advantages:

- It represents another kind of smoothing operator. To a certain extent a β schedule can be thought of in the same way as Graduated Non-Convexity [9] since we are starting with a strictly convex function when β is the minimum edge strength value, $min_{(i,j)\in\mathcal{E}}\Phi_{i,j}$, and all pixels are then separated into their own regions by the model. By slowly increasing $\beta_k \longrightarrow \beta$, the function assumes increasingly its β-dependent non-convex shape. However, whereas in GNC we can recover the original non-convex function by reducing the convexity parameter to 1, in a β schedule we must be careful not to choose a β value which is too high since this could lead to poor graph partition results.

- GMs allow the grouping of pixels and regions which are very close together with respect to the similarity criterion. Pixels or regions that are very similar can then create larger regions at higher levels. Next, the creation of ever larger regions avoids the problems described in the previous section as well as in Figure 5.7 and thus discourages region spilling. It is easy to show that at the finest level (graph $\mathcal{G}^{(0)}$), edges obtained with β_b contain all edges β_j for all $b < j$. However, due to boundary geometries, the same cannot be said of graphs $\mathcal{G}^{(s)}$ for $s > 0$ since region spilling can still occur depending on the schedule for β.

  A slower schedule with a finer grading of β should have a lower likelihood of causing edge spilling while a faster schedule will have a higher likelihood. The reason for this is that regions will have more time to form as β is slowly increased which will allow them to grow on both sides of an edge as in Figure 5.7. If the β schedule is not fine enough, then large regions might not be able to form on both sides of an edge and there would not be enough *structure* to ensure that the Φ for the gap in the edge is averaged into the other Φ values. Note that two regions separated only by one or two

pixels (even for a high β) would spill into one another as there wound not be enough edge length to build regions on both sides of the edge. The proof for an optimal β schedule would be an interesting future study.

This procedure can also be compared to the watershed and repeated waterfall algorithms [2]. By increasing β, we are essentially increasing the level of the liquid "flooding" the image and making the catchment basins ever deeper. Instead of stopping at the highest peak (the criterion for the watershed transform), our algorithm stops when the stationary conditional probability has been found. Each time the level of the liquid is increased, lower level dams are flooded and thus only the highest dams remain. At the final level, all but the highest dams/ridges are left. One important feature distinguishes our algorithm from the watershed transform. Namely, when using the Potts model in a nested aggregation framework, the edge weights are *averaged* over the length of the edge while in the watershed a single edge gap which is lower than the threshold (*e.g.*, β) causes region-to-region spilling.

Computational requirements are going to increase with the β schedule since we will be converging to a new model for each of the chosen values in the β schedule. However, this increase will be tempered by the nested aggregation framework since after running the first model on the full image, subsequent models should run on considerably smaller graphs (depending on the initially chosen β). One must be careful not to choose an initial β_0 that is so small that nodes in $\mathcal{G}^{(0)}$ do not want to merge. In any event, the decrease in the graph size will not be as fast as in the single model case and could be very slow for long β schedules.

Note that in the limit the finest possible grading of a β schedule would be given by the sorted list of unique Φ values in the graph $\mathcal{G}^{(s)}$. There are at most $4 \cdot N$ of those edges. The computational complexity at this extreme graduated model would be at best $O(N^2)$. Using this particular β schedule results in the Highest Confidence First algorithm (cf. Section 2.8). Stochastic Nested Aggregation with Graduated Models is then a generalization of HCF.

Finally, we modify the stochastic nested aggregation graph partition algorithm found in Algorithm 7 to include graduated models in order to prevent region-to-region spilling. The updated method is presented in Algorithm 8. We will refer to this as the Stochastic

Algorithm 8 SNA Graph Partition Algorithm with Graduated Models

Split the image by assigning random labels to all nodes $\{v_i^{(0)}\}$ or obtaining preprocessed atomic regions $\{V_j^{(0)}\}$
for $\beta_t = \{\beta_0, \ldots, \beta_b\}$ **do**
  for $s = 0, \ldots, s_{max}$ (from finest to coarsest) where s_{max} is the dynamically-determined final level in the hierarchy **do**
    Relabel images according to Algorithm 6.
    repeat
      for $v_i^{(s)}, \forall i$ **do**
        Minimize the energy in model (5.11)
        Update the node's label based on Gibbs sampling Algorithm 1
      end for
      Update T according to the desired schedule (2.24)
    until $|\mathcal{V}^{(s)}| = |\mathcal{V}^{(s+1)}|$
    Apply transition equations (5.12) and (5.13) to transition from level s to $s+1$
  end for
  Update $\beta_{i,j}$ and $\Phi_{i,j}$ based on the next $beta_{t+1}$
end for

Nested Aggregation Graduated Models (SNA-GM) framework. If only one β value is desired then the β schedule can be disregarded and the algorithm reverts to the one shown in Algorithm 7. The algorithm is similarly divided into two parts: a trivial image splitting part in the first step, and a region merging part in subsequent steps.

5.3.8 Discussion

In summary, potentially three different types of smoothing are at work when using stochastic nested aggregation on the Potts model: a bottom-up region aggregation (*i.e.*, from the original non-convex function to a convex one), a top-down smoothing in the form of the simulated annealing T parameter schedule, and another top-down smoothing using a β schedule (*i.e.*, from a convex function to the original non-convex one). These three processes interact in order to allow the solution to converge close to the global optimum. The

bottom-up process is intrinsically part of the framework. Both top-down processes are model- and optimization algorithm-dependent. Eight different algorithms can be derived from these three interactive processes.

Iterated Conditional Modes or ICM is a strictly local optimization method. It is only suitable for optimizing convex functions or non-convex functions starting with very good initial conditions. It is generally very fast but this advantage is outweighed by its high dependence on a good initialization.

Simulated Annealing or SA improves on ICM by allowing for convergence from any initial conditions. SA is ergodic and reversible in that any point in the function can be reached from any other point. However, SA has very high computational complexity and therefore is not practical. To make SA more practical a set of parameters is used (a schedule) that does not guarantee global convergence yet allows SA to escape most local minima. However, the complexity of these more practical versions of SA is still too high.

SNA-ICM applies the nested aggregation framework to ICM-based optimization allowing the breaking of label deadlocks and thus escaping local minima with each progression in nested levels. This simple change makes ICM a very powerful global greedy optimization method for graph partition problems. However, in some cases deep local minima far from the optimum are reached (*e.g.*, some forms of region-to-region spilling).

SNA-SA allows SA to converge considerably faster to the conditional probability distribution function than the flat-field version making SA a viable stochastic alternative to deterministic approaches such us ICM.

SNA-GM-ICM benefits from the nested aggregation's framework ability to avoid most local minima including some cases of region-to-region spilling. The algorithm benefits from the combination of the high speed of ICM and the acceleration due to SNA. The model parameter relaxation due to the Graduated Models strategy further enhances the convergence properties of SNA-GM-ICM by allowing the optimization algorithm to converge *gradually* close to the non-convex function's global optimum. This algorithm would be less expensive than standard GM-ICM (*i.e.*, no hierarchies) and more expensive than SNA-ICM. When the β schedule corresponds to a sorted list of all edge strengths in the graph (from lowest to highest), SNA-GM-ICM corresponds to Highest Confidence First [25]. Therefore, SNA-GM-ICM is a generalization of HCF.

SNA-GM-SA might allow the optimization process to escape some particularly deep local minima that would otherwise not be escapable. Given the same β schedule, the SNA-GM-SA would be usually much more expensive than SNA-GM-ICM due to the T schedule.

For completeness, we include the description of two algorithms based on graduated models that are not nested and that will not be tested in this thesis.

GM-ICM is a model relaxation method which allows a deterministic algorithm to get close to the global optimum of a function through transforming the energy function into a convex function and then gradually relaxing the model parameters to obtain a solution to the desired non-convex function. A very fine grading of this parameter relaxation is necessary to obtain a result close to the global minimum; however, this requires a very high computational complexity. As the grading is made coarser, the ability to reach a point close to the global optimum is compromised. Furthermore, the algorithm might run into a deep local minimum as is the case with SNA-ICM.

GM-SA algorithm would allow GM to avoid some local minima created through a non-optimal parameter grading since a very fine model parameter grading might be to computationally expensive. However, some particularly deep local minima might still not be avoided.

The most beneficial of those eight algorithms are the four based on SNA as they allow fast convergence and enable the practical study of Markov Random Fields for image segmentation-type applications as well as other problems where computational efficiency is necessary to obtain practical results. Furthermore, SNA gives us a framework which allows the use of other model types at different levels in the hierarchy. One of these alternative models will be discussed next.

5.4 Region-Based Characteristics: The Mean Model

Up to this point, we have assumed that the dissimilarity of two regions is computed based on a pixel-to-pixel distance measure which in image processing would be a simple edge gradient computed using the Euclidean distance (4.10) or some other distance measure. Since a 4-neighborhood is used, this gradient is usually computed in the vertical and horizontal

directions for all adjacent pixels using $\Phi(v_{i,j}, v_{i,j+1})$ and $\Phi(v_{i,j}, v_{i+1,j})$ respectively. These pairwise criteria are then used in the assessment of similarity and dissimilarity of adjacent pixels and at higher levels, of adjacent regions. This assumes that the image is a piecewise constant function with large constant homogenous areas punctuated by sharp transitions or edges. We assume that the essential information for computing the difference between two regions lies in the transition or edge areas and not in any other part of the image.

When using pairwise distance measures, all computations are based on comparisons of two variables or nodes in a graph. The main implication of this computation for SNA is that on any particular level of the hierarchy some nodes that are far apart spatially could be given the same label due to identically labelled intermediate nodes linking those two nodes. This is especially problematic at the finest or pixel level where many nodes in the same vicinity can be assigned the same label forming small regions or patches. However, a major advantage of pairwise computations is that we do not have to base our comparisons on more than two quantities (*i.e.*, two pixels or regions). Furthermore, using specifically the pixel-to-pixel gradient allows us to solve the same model at higher levels in the hierarchy given that $\mathcal{G}^{(s)}$ are equivalent. It is important to note that even when computing distances between groups of edge pixels for $\mathcal{G}^{(s)}$ with levels $s > 0$, the distances are all pairwise since at higher levels of the hierarchy a single node represents many underlying pixels.

However, if a non-local relationship between nodes is used, the labelling of those nodes might end up being closer to what we would consider a "correct" labelling. Non-local relationships are defined by pixel neighborhoods that are larger than the 4-neighborhoods used in the Potts model at the finest scale. For example, the relationships between regions at higher levels in the SNA could be considered non-local if the number of pixel edges is greater than one.

One very attractive extension of the notion of graph partitioning involves using the finest scale model to create small regions with one Potts model (5.11) which can then be aggregated using different and more perceptually correct Potts models at the higher levels in the hierarchy. As the levels in the hierarchy increase and the regions become larger, one could envision using two or three different models to obtain a partition with each set of subsequent levels using a model appropriate for the given scale.

In image processing, this would be necessary as the *edge*-based region formation model

(especially without a β schedule) is prone to region spilling. One way to mitigate this effect might be to compare regions according to their means. Computing the mean of a region averages all the pixel values within that region including the edge pixels making it difficult to merge two different adjacent regions connected by a slowly varying gradient[8]. The mean-based model assumes that the pixels in the inner part of the region are as important to assessing region similarity as edge pixels. The segmentation model still insures that the image is a piecewise constant function; however, the nature of this function is now altered to include all pixels in the image (not just edge pixels).

The mean-based model is exactly the same in appearance as model (5.11). However, the distance computations between regions are now carried out using means. This introduces a subtle shift in the way larger regions form. At higher levels of pixel organization edges do not matter as much as they do when forming small region patches or blobs. The region mean (and in the case of vector angle the first principal component or principal direction) is able to describe regions in a fuller manner by taking into account all pixels in the region and not just the edge pixels. If region spilling occurs at the finest level $s = 0$, then no region-to-region spilling avoidance strategy will work. To mitigate the errors at the finest level by having a very low initial β that restricts merging to the closest adjacent pixels.

In this mean model, the transitions take on a different form for $\Phi_{r,r'}^{(s)}$ (they remain identical to the edge-based model for $\beta_{r,r'}^{(s)}$). Given that the model no longer cares about individual pixels but groups of pixels, the transitions need to account for changes in the number of pixels in a region, as well as the change in region mean values. The new transition equations for $\Phi_{r,r'}^{(s)}$ are defined by

$$m_r^{(s)} = \sum_{i \in V_r^{(s)}} m_i^{(s-1)} \frac{n_i^{(s-1)}}{n_r^{(s)}} \tag{5.14}$$

$$n_r^{(s)} = \sum_{i \in V_r^{(s)}} n_i^{(s-1)} \tag{5.15}$$

$$\Phi_{r,r'}^{(s)} = \Phi(m_r^{(s)}, m_{r'}^{(s)}) \tag{5.16}$$

where at any level s, $n_r^{(s)}$ are the number of pixels in a given region r and $m_r^{(s)}$ represent the region means.

[8]Preliminary work was presented in [156].

The Potts *mean* model is related to the Potts *edge* model and is another possible generalization of the finest scale Potts *edge* model as each initial node or pixel can be considered to be a region with just one node whose mean is its own vector (*e.g.*, scalar for grayscale images or vector for color or multispectral images). However, the energy models $U^{(s+1)}$ and $U^{(s)}$ no longer represent exactly the same models as was the case for the energy model due to the non-local interaction built into the region means. Because of this, a β schedule (however short) should be used in order to allow pixels and then regions to be grouped at a slower rate.

As before the image segmentation algorithm is divided into two parts: a trivial image splitting part in the first step and a region merging part in subsequent iterations. The only difference with Algorithm 7 is that the region means are updated after the iterations on a particular level s have been completed.

Nested aggregation will converge to a local or global minimum for the last graph $\mathcal{G}^{(s)}$ just like it does for the edge model. However, this will not be the same minimum as in $U^{(0)}$ due to the nonlinear transformations from model $U^{(0)}$ to $U^{(s_{max})}$. Due to this non-locality property, convergence to a good local minimum in $U^{(s_{max})}$ hinges on a delicate balance of $\beta^{(s)}$ and $\Phi^{(s)}$ at each level. In this thesis, we ensure this balance through the use of a β schedule.

Other more creative models can also be used where the relationships between the regions are more complex combinations of mean-based and edge-based calculations. Furthermore, more complex distance measures such as those used in edge detection can be used instead of the simple metrics examined here.

5.5 Results

Several images are shown in this section to demonstrate how nested aggregation performs in an image segmentation task in conjunction with a Potts model. We use here a simple color-based non-texture model to determine similarity between pixels: the vector angle measure (4.12).

Remember that because the dot product between the vectors is divided by the magnitude of the vectors, the distance is naturally intensity invariant with respect to the Dichro-

matic Reflection Model [148]. In addition, one of the problems with vector angle is that it produces very "noisy" results for vectors with small magnitudes [127, 150]. Furthermore, any areas without chromatic information will be grouped together.

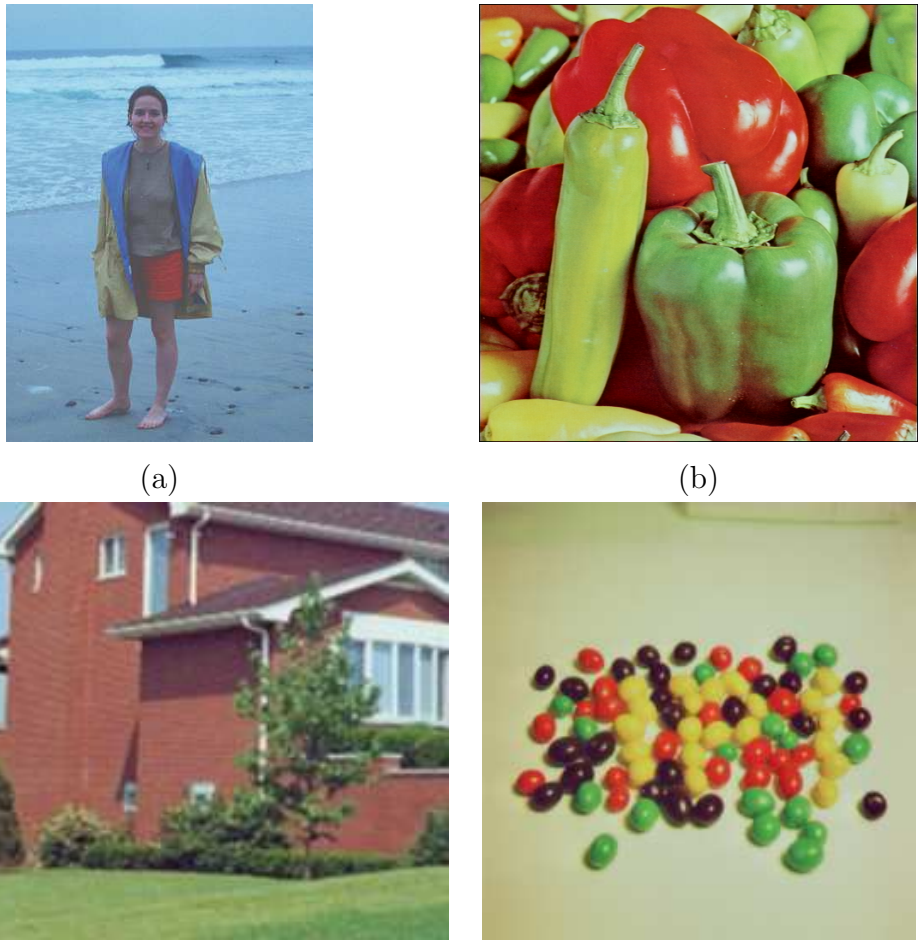
We will examine results for Stochastic Nested Aggregation with Graduated Models and ICM (SNA-GM-ICM) and with SA (SNA-GM-SA) using both the edge-based and mean-based Potts models. Figure 5.17 shows the images which will be tested. In each section below, we will first examine results in detail for Figure 5.17(a). Other images will be briefly examined at the end of each section. All segmentations are carried out for $K = 10$ labels except where indicated together with a relabelling process applied at the end of each level (example of not applying relabelling will be shown shortly in order to motivate the need for it). Parameters for the ICM and SA algorithms used in the optimization are given together with each result (some parameters vary from image to image). In general, the number of nested levels is unbounded and varies from image to image and algorithm to algorithm.

The assessment of results will be made from a qualitative rather than quantitative point of view. This is because quantitative segmentation evaluation would need to be based on an ideal segmentation of an image which is difficult to determine without an application at hand. When phase unwrapping will be considered in Chapter 8, some quantitative results based on the computed digital elevation model will be provided.

5.5.1 Potts Edge Model

In this section, we show results for the Potts "edge" model. The SNA-GM-ICM algorithm was run with two iterations at each level in a similar fashion to SNA-ICM (see Section 5.3.6). Figure 5.18 shows results for a short β schedule, $\beta = \{0.0010, 0.0020, 0.0030, 0.0040, 0.0045\}$ which ran on average 60 seconds. The SNA-GM-ICM produces much more desirable results on average than SNA-ICM. Many local minima are avoided thanks to this graduated models strategy.

However, since we are using ICM, the results do not always converge to a desirable minimum. They still depend very much on the initial conditions; namely, the initial labelling. Several results for the exact same set of parameters (but different initial labelling) are shown in Figure 5.19 and indicate that initial conditions affect the performance of



(a) (b)

(c) (d)

Figure 5.17: Test images: (a) woman (image size: 310×442), (b) peppers (image size: 512×512), (c) house (image size: 255×192) and (d) jelly beans (image size: 256×256). All images have been chosen such that few texture appear in them since this thesis does not address the issue of texture segmentation and aims to illustrate function minimization using stochastic nested aggregation. Many texture descriptors exist in the literature [4, 22, 32, 75, 106, 167].

SNA-GM-ICM to some extent.

Figure 5.20 shows that image segmentation results improve as we increase the gradation of the β schedule. A finer schedule such as $\beta = \{0.0004, 0.0006, \ldots, 0.0058, 0.0060\}$ generates considerably better results than those shown in Figure 5.19. Thus, initial conditions seem to have a lesser effect when a longer β schedule is used. However, with this new schedule the SNA-GM-ICM now takes on average 210 seconds to complete. Note that increasing the final β to 0.0060 does not seem to have affected the result significantly (*e.g.*, create widespread region merging) whereas already a $\beta = 0.0055$ merged most of the regions as shown in Figure 5.14. Therefore, SNA-GM-ICM seems to be also more robust with respect to the model (*i.e.*, β) selection due to the gradual formation of regions.

Figure 5.21 shows examples of image segmentation for a small number of labels with no relabelling. The effects of region spilling due to the labelling critical slowing down effect can be seen in all images and are quite severe. Relabelling, shown for example in Figure 5.19, has been very effective at mitigating the effects of critical slowing down. It is also computationally much more efficient than increasing the number of labels. Increasing the number of labels (*e.g.*, to $K = 100$) and then computing energies based only on neighboring labels has an effect on image segmentation. Figure 5.22 shows three individual results which contain region spilling errors or unfinished segmentation (background is not one separate region but two). They are perhaps slightly better than results shown in Figure 5.19. Furthermore, the computational cost is high with an average of 606000 total site visits, 275 seconds running time and 18 levels. Therefore, although results for using a large number of labels might be comparable (and slightly better at best) than those obtained using relabelling, due to the computational cost of the procedure, we will use relabelling.

SNA-GM-SA was also run using the above β schedules with the results presented in Figure 5.23 using a long temperature schedule in SA ($T_o = 5$, $\tau = 0.8$ and with 29 iterations followed by 3 iterations at $T = 0$). Results show considerable improvement over SNA-GM-ICM especially in the case of the longer β schedule. However, the computational cost of obtaining those results is significantly higher. For the shorter β schedule the average running time was 5.5 minutes (16 levels and 8.8 million site visits) while for the longer schedule it took on average 16.5 minutes (60 levels and 25.5 million site visits) to converge to the

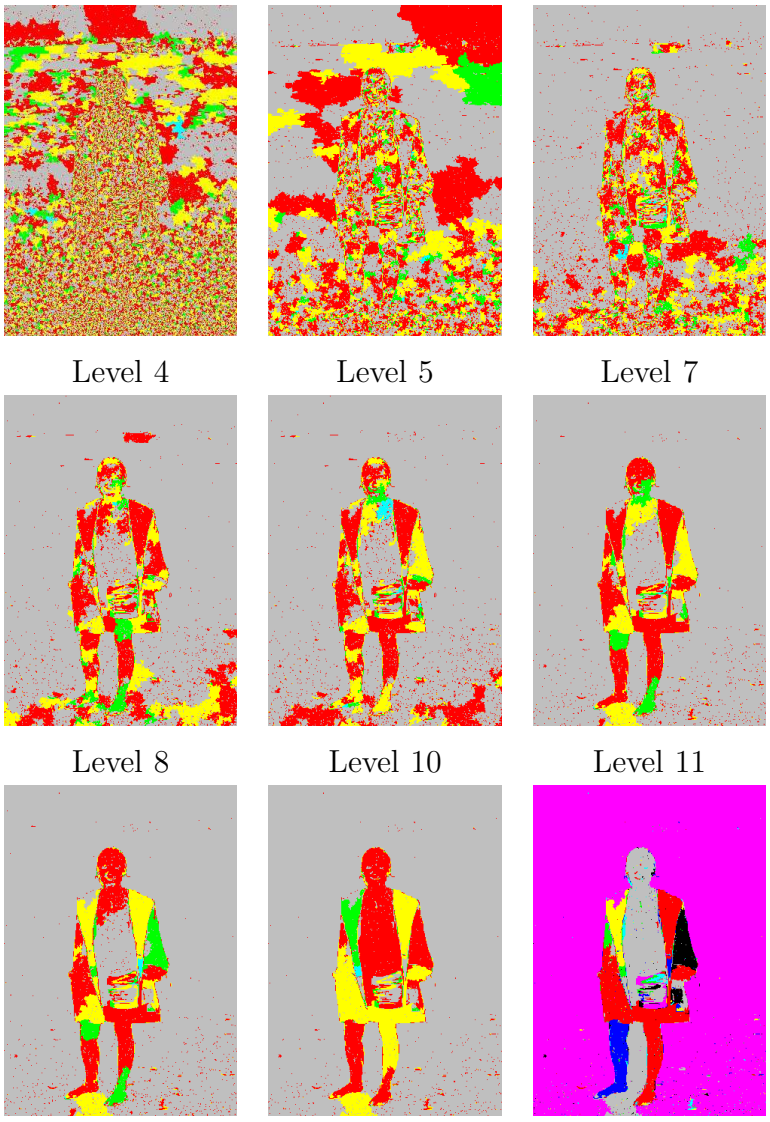


Figure 5.18: Detailed example of SNA-GM-ICM with the vector angle distance measure in Potts model (5.11). A $\beta = \{0.0010, 0.0020, 0.0030, 0.0040, 0.0045\}$ schedule was used. The transitions between β values resulted in significant numbers of pixels being grouped together: *e.g.*, Level 4 was the last level for $\beta = 0.0010$ while Level 5 was the first level for $\beta = 0.0020$.

SNA-GM-ICM SNA-GM-ICM SNA-GM-ICM

Figure 5.19: Initial conditions still affect the performance of ICM in SNA-GM-ICM when using a short β schedule. The same parameters were used as for results in Figure 5.18. Three independent results are shown. The computational cost was on average 60 seconds.

SNA-GM-ICM SNA-GM-ICM SNA-GM-ICM

Figure 5.20: Initial conditions still affect the performance of ICM in SNA-GM-ICM even when a longer β schedule is chosen: $\beta = \{0.0004, 0.0006, \cdots, 0.0058, 0.0060\}$. However, the results are better than in Figure 5.19. This could be attributed to the gradual creation of regions with a longer edge next to region-spilling pixels thus preventing region spilling from occurring. Three independent results are shown.

SNA-GM-ICM SNA-GM-ICM SNA-GM-ICM

Figure 5.21: A small number of labels ($K = 10$) was applied to the image segmentation task using SNA-GM-ICM with the short β schedule. Results are worse than when using $K = 10$ with relabelling shown in Figure 5.19. Region spilling is severe and present in all examples. The computational cost was on average 60 seconds.

label configuration shown. This indicates that SA should only be used when considerable processing power is available. One way to reduce this computational burden is to change the model from an edge-based one to a mean-based one. We will show in Section 5.5.3 that we can achieve similar results to SNA-GM-SA using SNA-GM-ICM with an order of magnitude lower computational requirement thanks to changing the model. Otherwise it would be more reasonable to use SNA-GM-ICM and try to mitigate initial conditions considerations by using a very low β to obtain a better set of initial atomic regions.

It is interesting to note that the results for the long β schedule for both ICM and SA are very similar (compare Figure 5.20 and bottom row of Figure 5.23). This is most likely due to the small number of nodes being merged at each level due to the fine grading of the graduated models. For images where region spilling problems are more accentuated, SNA-GM-ICM might not perform as well as SNA-GM-SA. This is left as a future exercise.

Figures 5.24, 5.25 and 5.26 summarize results for the peppers, house and jelly beans images. Two or three final independent results are provided for each of these images (and in each case for SNA-GM-ICM and SNA-GM-SA) in order for the to evaluate the range of

SNA-GM-ICM SNA-GM-ICM SNA-GM-ICM

Figure 5.22: A large number of labels ($K = 100$) applied to the image segmentation task using SNA-GM-ICM with the short β schedule and no relabelling. Results are comparable to the relabelling cases (see results in Figure 5.19), but none are as good as the results for SNA-GM-ICM with a long β schedule (see Figure 5.20). There is region spilling in the first two examples and the third example has a large region that is separate from the main background region.

β^1 schedule β^1 schedule β^1 schedule

β^2 schedule β^2 schedule β^2 schedule

Figure 5.23: Detailed example of SNA-GM-SA with the vector angle distance measure in Potts model (5.11). A (short) schedule $\beta^1 = \{0.0010, 0.0020, 0.0030, 0.0040, 0.0045\}$ was used for the top row while in the bottom row the results are for (long) schedule $\beta^2 = \{0.0004, 0.0006, \cdots, 0.0058, 0.0060\}$. The transitions between β values resulted in significant numbers of pixels being grouped together producing a desirable final result. SA was also effective at avoiding many local minima evident in top row results (as compared with results in Figure 5.19). Three independent results are shown for each of the schedules.

the obtained local minima.

The use of a stochastic optimizer such as SA appears to give on average better results. However, they are not stunningly better as in the case of Figures 5.12. It would appear that applying SNA-GM has narrowed the differences between ICM and SA considerably. SA-based algorithms are still better than ICM due to ICM's dependence on initial conditions as shown in Figures 5.19 and 5.20. SA is able to avoid this dependence by going to a higher energy state and in effect finding a better set of initial conditions. However, SA with the exponential schedule still does not guarantee convergence to a global minimum (and to some extent is still dependent on initial conditions since the T schedule does not decrease slowly enough). However, the SA- and ICM-based algorithms used here are able to obtain very good local minima and show that graph partitioning of a finest scale energy model can be accomplished in a fast and efficient way. Our approach is much faster than the cluster sampling approach [4] where the authors obtain a 400-fold speed-up and compares favorably in speed with the fastest graph cuts approaches [11].

5.5.2 Potts Mean Model

The pixel dissimilarity criterion Φ was chosen to be the vector angle measure following [148] which is invariant to illumination intensity and shading. Furthermore, the region mean was replaced by its vector angle analogue, namely, the first principal component of the covariance matrix of the pixel vectors as was done in the MPC [34, 148]. This value represents the most prevalent vector direction in the region.

The results in Figure 5.27 show image segmentation using the mean model with optimization done using SNA-GM-ICM. Region spilling is much more prevalent than in the edge-based Potts model case. The mean Potts model exhibits more region spilling due to using means to represent regions instead of edge pixels. This can be easily explained by the way the algorithm works. At the first level, the pixels are gathered into some small regions. Then the means of those regions are computed and compared. However, many pixels might not have had the chance in those first two iterations to coalesce into small regions. Therefore, these pixels are merged into adjacent regions in similar manner to the process a region growing algorithm would use. Since the mean of each of those regions changes with every level due to the inclusion of new pixels, we have now a lot more variability in the

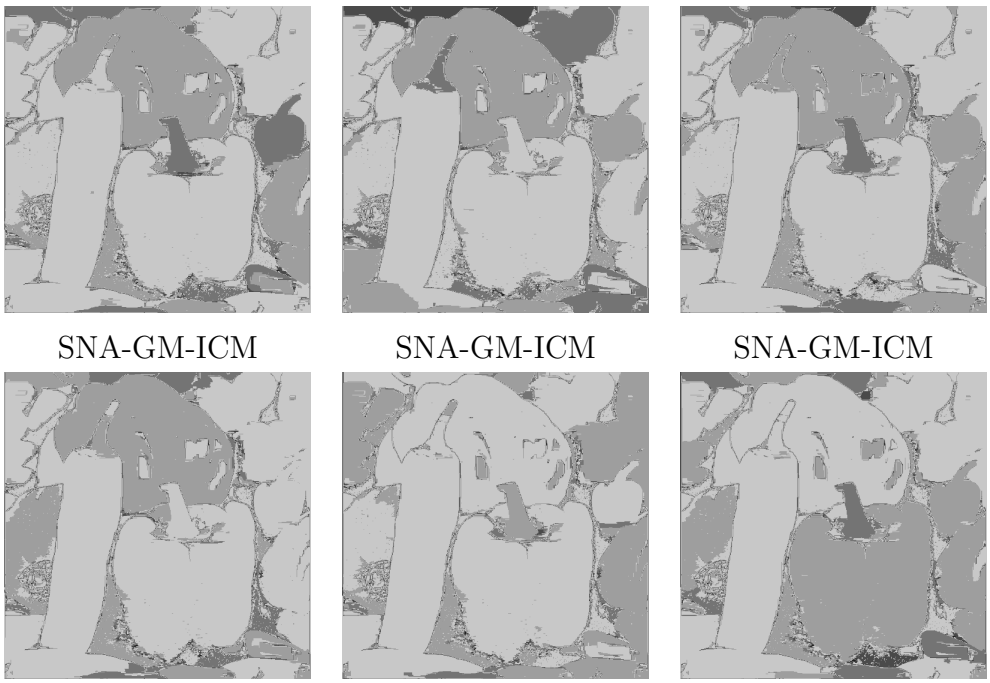


Figure 5.24: Segmentation results for the peppers image using SNA-GM-ICM and SNA-GM-SA with $\beta = \{0.0010, 0.0020, 0.0030, 0.0040, 0.0045\}$. SNA-GM-SA was run with $T_o = 1$, $\tau = 0.4$ and with 7 iterations followed by 3 iterations at $T = 0$. The peppers image gives very stable results for either algorithm. Some region spilling can be observed; however, it is very limited in extent. All the large peppers in the forefront are segmented in a desired fashion. The average statistics for the SNA-GM-ICM algorithm are 140 seconds running time, 19 levels and 1.25 million site visits while for the SA with the given T schedule they are 350 seconds, 17 levels, and 5.9 million site visits. The SNA-GM-SA was considerably more expensive for the slightly better qualitative performance. Note that the large highlight parts of the green and red peppers are still segmented as different objects. Three independent results are shown for each algorithm.

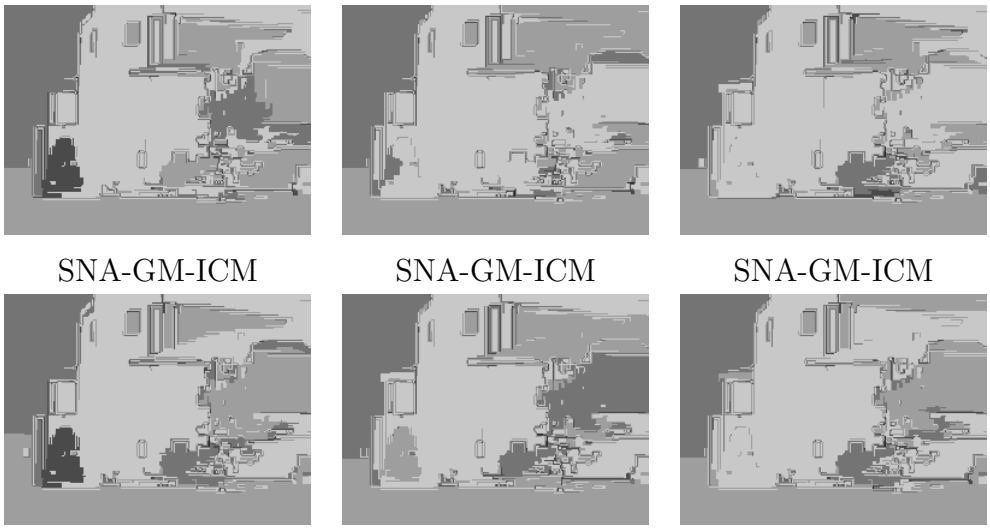


Figure 5.25: Segmentation results for the house image using SNA-GM algorithms with $\beta = \{0.0001, 0.0002, 0.0004, 0.0006, 0.0008, 0.0010, 0.0011\}$. SNA-GM-SA was run with $T_o = 1$, $\tau = 0.4$ and with 7 iterations followed by 3 iterations at $T = 0$. The house image shows more spilling (than the peppers or woman images) especially in the case of SNA-GM-ICM algorithm. Note that the image was filtered using a Gaussian 5×5 filter with $\sigma = 0.8$. Three independent results are shown for each algorithm.

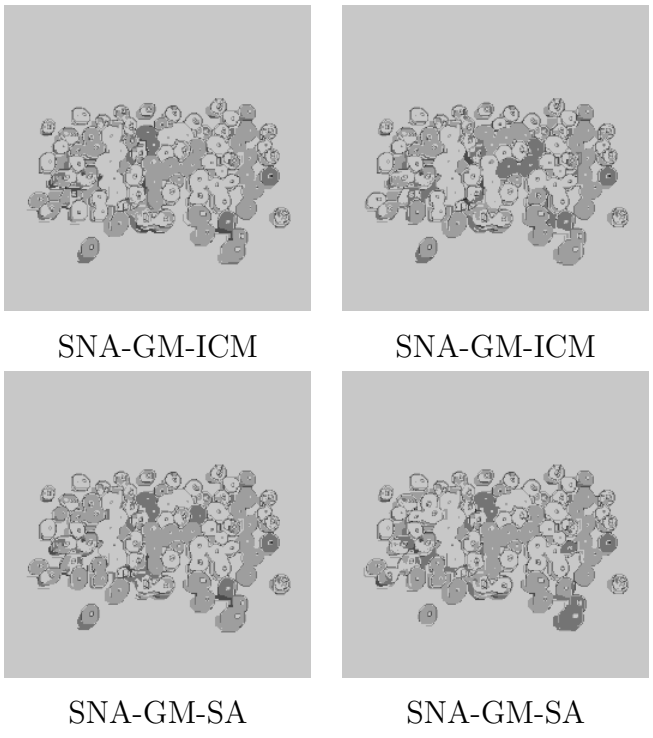


Figure 5.26: Segmentation results for jelly beans image using SNA-GM-ICM and SNA-GM-SA: $\beta = \{0.0010, 0.0020, 0.0030, 0.0040, 0.0045\}$. Highlight regions are visible throughout the segmented image. Observe that in all cases the dark beans are very noisy due to the vector angle measure. Two independent results are provided for each algorithm.

means which results in highly undesirable segmentations.

Once pixels coalesce into regions they are represented by a mean. The distance calculation is based on non-local values and, therefore, the random field no longer exactly represents model (5.11). This influence of causality (*i.e.*, the past history of included pixels will determine which pixels are added next) is one of the main problems associated with region growing and has been one of the handicaps we have avoided until now. The results in [156] looked promising but they were in fact based on very carefully chosen parameters. Slight changes in parameters would cause a great deal of region spilling as is evidenced by the results in this section.

The unreliability of results using the mean Potts model is illustrated further in Figures 5.28 and 5.29 where segmentations of two other test images are shown. These results show clearly that the mean Potts model is clearly not appropriate for image segmentation.

5.5.3 Mixed Models

The main problem with the mean model seems to be a "too rapid" transition to computing distances based on means. It would be perhaps more desirable to first compute atomic regions based on the edge model with a conservative β. We can then compute a region mean or prototype for these regions in order to guard against region-to-region spilling due to smooth edge transitions (that would spill at a higher β). Therefore, in a final experiment, we combine the Potts edge model with the Potts mean model by applying the edge model first as a preprocessing step in order to obtain atomic regions. This operation is followed by the mean model with a very long β schedule. The transitions between β values are very gradual in order to allow only a few regions or pixels to join together at any one time.

The advantage of the edge model is in the grouping at the finest scale where local differences are much more important than regional or global ones. The mean model appears to be advantageous at higher scales since it is non-local. This set-up fits very well the SNA framework where we split the optimization problem into two subproblems: one to obtain small compact regions at a finer scale and the other to group those compact regions at the higher scales. This perceptual organization framework also agrees with other researchers who first create atomic regions in order to then group them into larger structures [4, 22, 119].

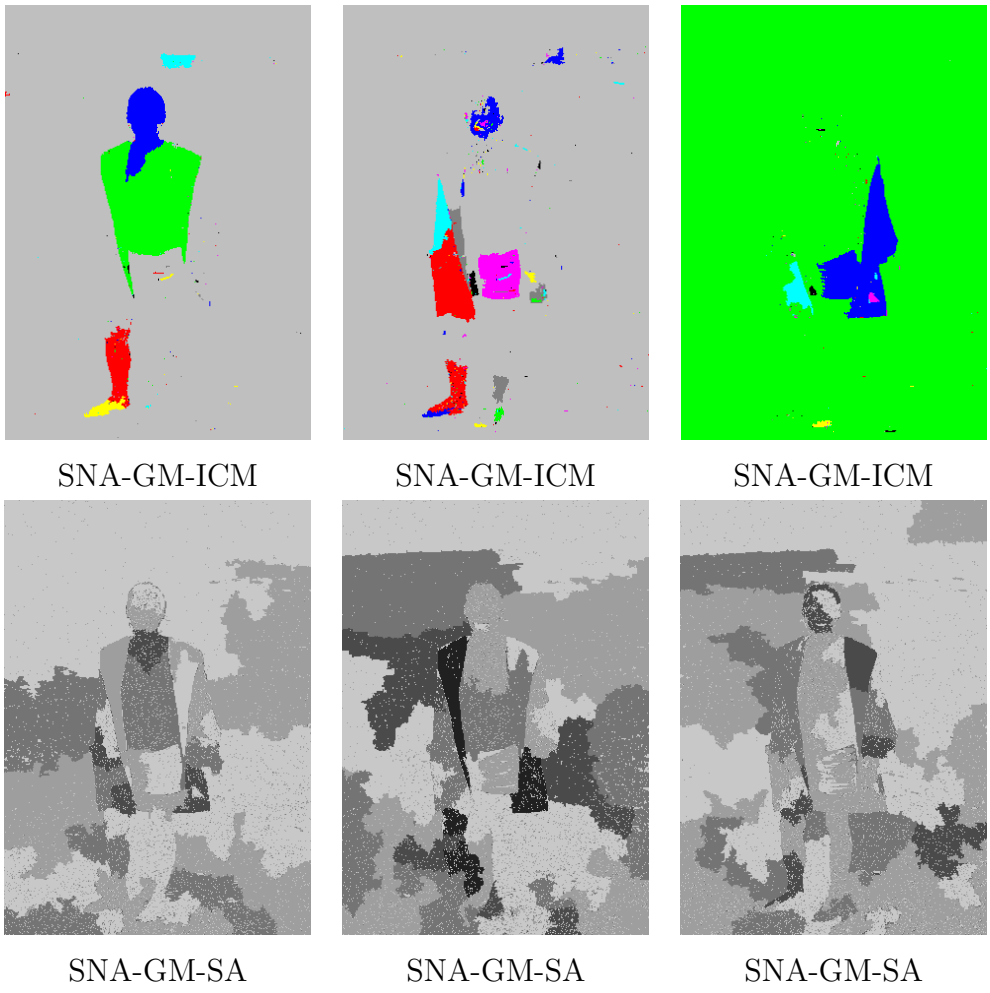


Figure 5.27: Examples of Results for the SNA-GM paradigm using the short schedule $\beta = \{0.0010, 0.0020, 0.0030, 0.0040, 0.0045\}$ for the mean Potts model. SNA-GM-SA was run with $T_o = 1$, $\tau = 0.4$ and with 7 iterations followed by 3 iterations at $T = 0$. Three independent results are shown for each algorithm.

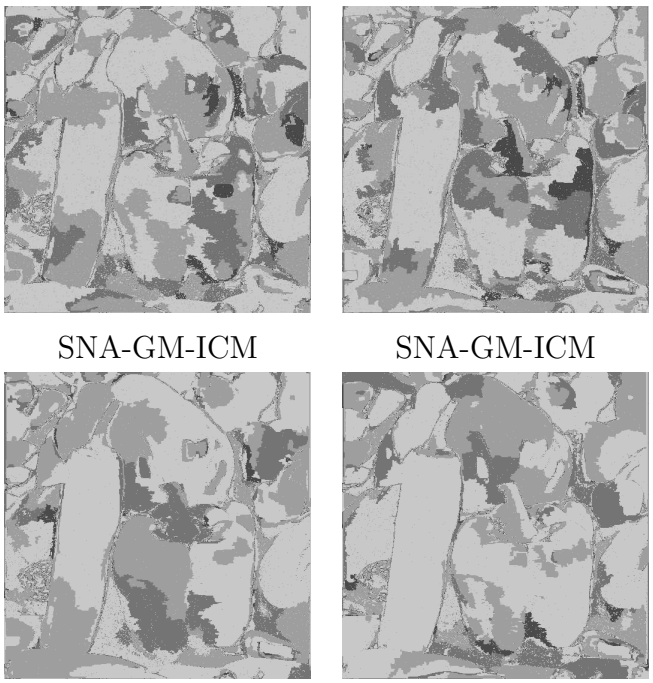


Figure 5.28: Segmentation results for the peppers image using the SNA-GM paradigm with $\beta = \{0.0010, 0.0020, 0.0030, 0.0040, 0.0045\}$. SNA-GM-SA was run with $T_o = 1$, $\tau = 0.4$ and with 7 iterations followed by 3 iterations at $T = 0$. The results show a lot of region spilling which did not occur for the edge model. The mean model took considerably more time to converge than the edge model especially due to the changing prototypes which caused new regions to be merged together at higher levels of the hierarchy. The average statistics for the SNA-GM-ICM algorithm are 405 seconds running time, 48 levels and 2.45 million site visits while for the SA with the given T schedule they are 740 seconds, 58 levels, and 12 million site visits. There does not seem to be any benefit (be it computational or from the point of view of desired result) from using SA in this case. Two independent results are shown for each algorithm.

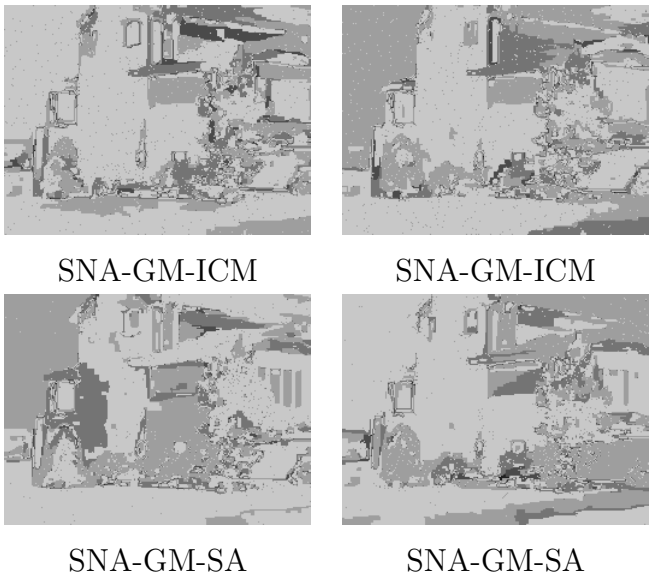


Figure 5.29: Segmentation results for the house image using the SNA-GM paradigm with $\beta = \{0.0001, 0.0002, 0.0004, 0.0006, 0.0008, 0.0010, 0.0011\}$. SNA-GM-SA was run with $T_o = 1$, $\tau = 0.4$ and with 7 iterations followed by 3 iterations at $T = 0$. The undesirable results are similar to those for other images. Two independent results are shown for each algorithm.

The nested aggregation framework lets us combine these two models seamlessly by carefully using a new β schedule, $\beta = \{0.0010, 0.0011, 0.0012, \cdots, 0.0044, 0.0045\}$. Thus, the edge model was run using $\beta = 0.0010$ which was then followed by a very slow β schedule for the mean model. A very slow schedule is implemented in order to allow the mean model to adapt slowly since the shape of the function being optimized changes with each modification in region prototypes. We postulate that if this change happens slowly enough the segmentation at level s will be a good starting point for the partitioning operation at level $s + 1$.

Figure 5.30 and Figure 5.31 show some results using SNA-GM-ICM and SNA-GM-SA respectively. The algorithm ran on average for $70 - 80$ levels taking on average 2 minutes to complete for SNA-GM-ICM case while taking on average 5.5 minutes (73 levels and 5.9 million site visits) for SNA-GM-SA with a short T schedule ($T_o = 1$, $\tau = 0.4$ with 6 iterations followed by 3 iterations at $T = 0$) and 13 minutes (73 levels and 19.5 million site visits) for the longer schedule ($T_o = 5$, $\tau = 0.8$ with 30 iterations followed by 3 iterations at $T = 0$). The results are remarkably stable for both the deterministic and stochastic versions of SNA-GM. This result confirms the need to create atomic regions when carrying out image segmentation.

Results obtained for the other images are shown in Figures 5.32, 5.33 and 5.34.

5.6 Summary

Top-down smoothing operations, the β and T schedules, are model- and optimization algorithm-dependent respectively while the bottom-up stochastic nested aggregation smoothing operation is model-dependent. Each of these different smoothing operations has significant implications for computational speed and the ability to converge near to the global optimum. There is a clear trade-off between computational speed and global convergence that is achieved based on how the β and T schedules are specified. In general, it is well known that a slower schedule for T will lead to better solutions than a faster schedule [48]. Also, a slower β schedule (*i.e.*, with a larger set of β values) will enable the algorithm more readily to escape local minima.

The fastest time and least convergence ability is given by $T = 0$ and a single β value

Figure 5.30: Examples of results for SNA-GM-ICM using a long β schedule and two models: several levels at the first β value using the Potts edge model followed by the remaining β values using the Potts mean model. Some areas exhibit minor region spilling; however, overall the results are very encouraging especially since they were obtained using ICM. Three independent results are shown.

(*i.e.*, ICM) while the slowest time and best convergence properties are given by a T schedule and a very finely graded β schedule (*i.e.*, SNA-GM-SA). Note that the best β schedule would correspond to a list of all the edge values Φ in the graph sorted in ascending order until a value of β that defines the model (this corresponds to the Highest Confidence First algorithm). In other words, if we have the finest possible grading of β so that no $\Phi_{i,j}$ is omitted, this would allow the algorithm to merge only those pixels/regions which should be merged and not any others. This would result in a deterministic algorithm such as HCF. If some values of $\Phi_{i,j}$ in the above generated sequence were omitted for the sake of decreasing the computational complexity, then a stochastic optimization algorithm would be needed since region-to-region spilling due to small gaps could now occur.

Stochastic nested aggregation has a positive impact on computational speed by providing geometric convergence through higher levels of the hierarchy. This new framework for hierarchical function optimization allows the grouping of any number of nodes into one node based on a well-defined global criterion (the energy model). Algorithm 7 is a generalization of many algorithms that exist in the literature (including ICM, SA, HCF,

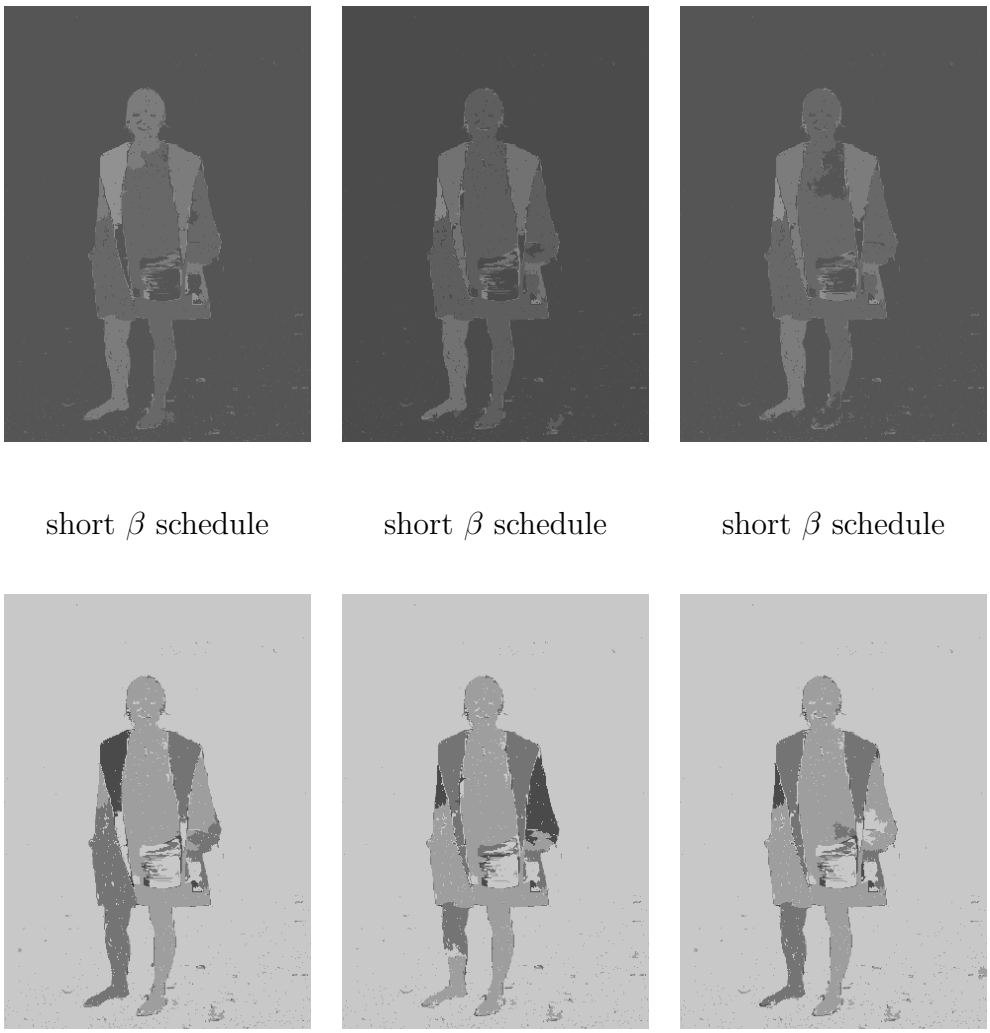


Figure 5.31: Examples of results for SNA-GM-SA using a short β schedule (top row) and long schedule (bottom row) together with two models: Potts edge model followed by the Potts mean model. Three independent results are shown for each schedule type.

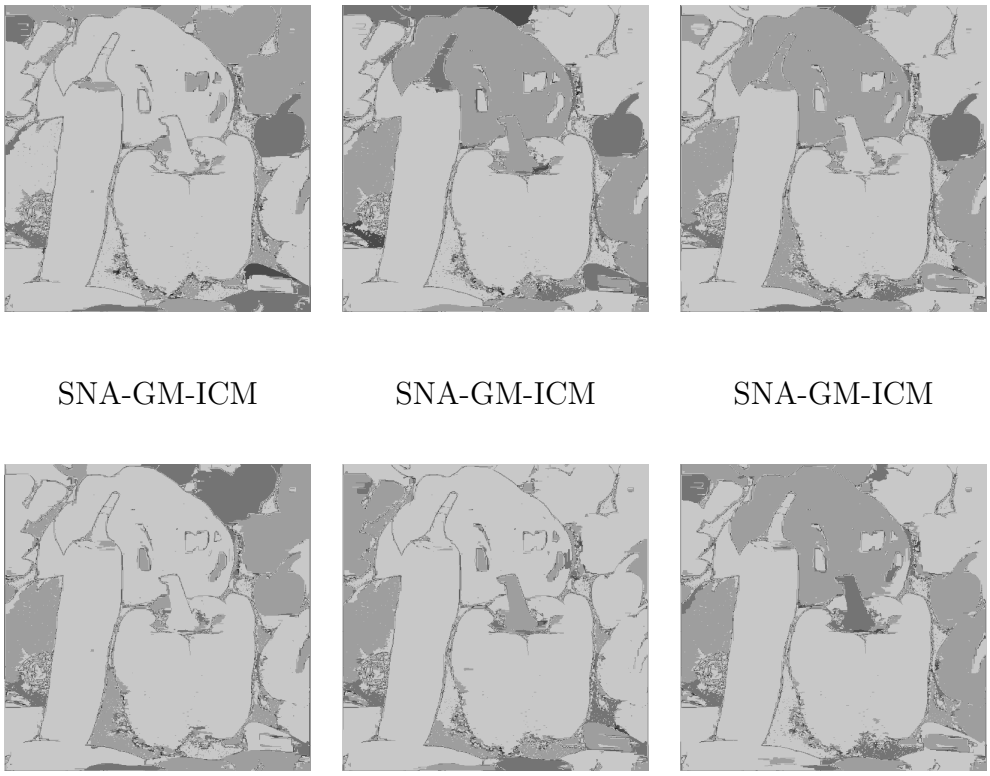


Figure 5.32: Examples of results for SNA-GM-ICM and SNA-GM-SA using a long β schedule and two models: Potts edge model and Potts mean model. We used a short temperature schedule for SNA-GM-SA. Three independent results are shown for each algorithm.

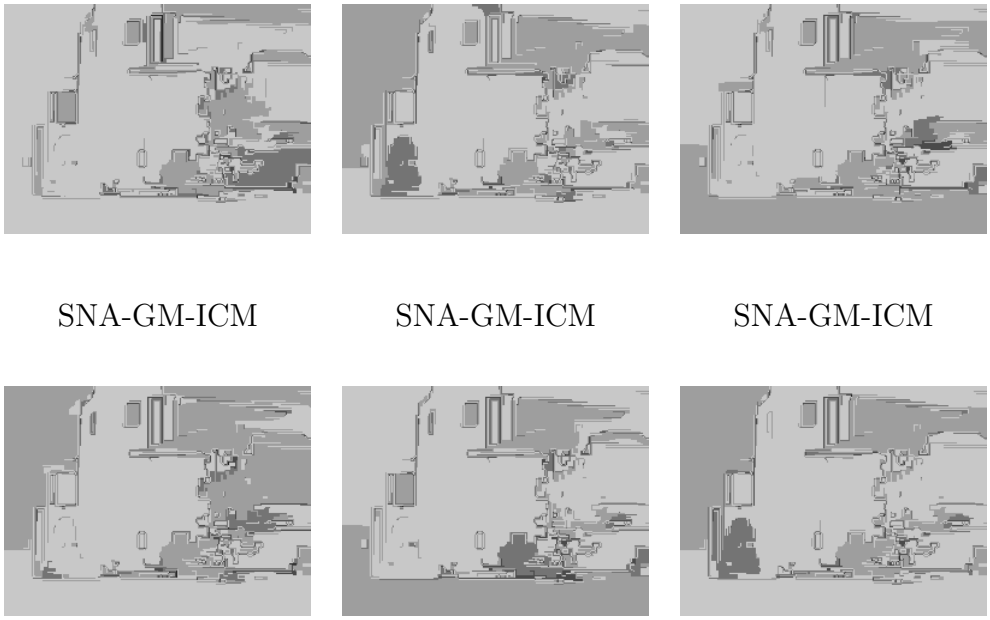


Figure 5.33: Examples of results for the SNA-GM-ICM and SNA-GM-SA algorithms using a long β schedule and two models: Potts edge model and Potts mean model. We used a short temperature schedule for SNA-GM-SA. Three independent results are shown for each algorithm.

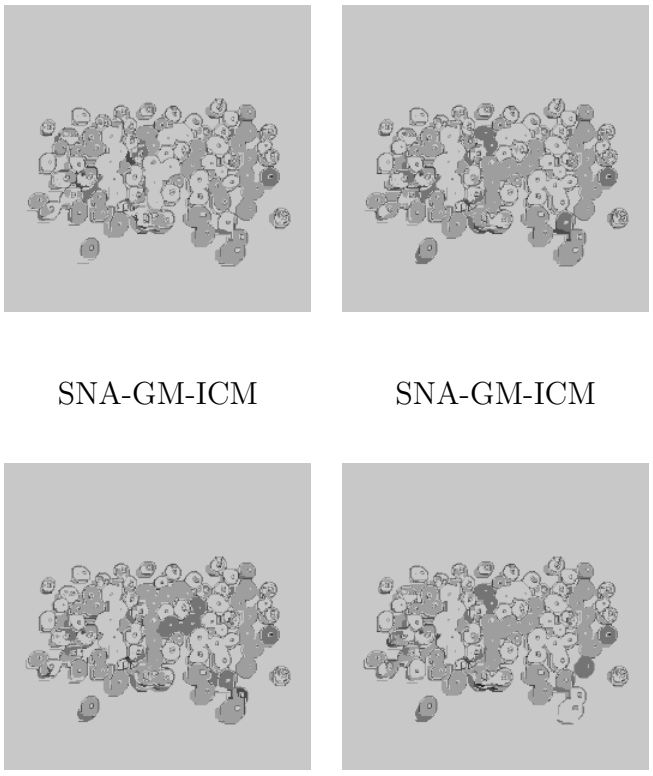


Figure 5.34: Examples of results for the SNA-GM-ICM and SNA-GM-SA algorithms using a long β schedule and two models: Potts edge model and Potts mean model. We used a short temperature schedule for SNA-GM-SA. Highlights are visible in all results. There is also a considerable amount of noise in segmented dark areas. Two independent results are shown for each algorithm.

and region growing).

Through the use of the stochastic nested aggregation framework, we are now able to use the ICM algorithm without needing to search for an optimal initial set of labels which in many cases is a suboptimal graph partitioning result obtained using another method (*i.e.*, oversegmentation in image processing terms). Stochastic nested aggregation is thus able to break labelling deadlocks which plague ICM and therefore allow the algorithm to escape many local minima. This enhancement makes ICM a viable alternative to SA for many problems where a global partitioning of a graph is necessary. In the case of SA, SNA is able to significantly accelerate its computational speed, therefore making SA a viable alternative to ICM (many researchers would not use SA due to its computational complexity). Even the smallest amount of processing at a higher temperature benefits results thus allowing the optimization using very fast temperature schedules.

SNA is also a framework which allows different models to be used at various levels of the hierarchy. We have shown that the mean Potts model was not very useful by itself; however, it proved to be a very effective add-on to the edge Potts model especially at levels in the hierarchy where atomic regions or partitions have already been formed.

Chapter 6

Pixel Comparison: Color Spaces and Metrics

The purpose of a metric in image segmentation is to be able to quantify with some meaning distances between pixels or regions. This chapter discusses statistics- and physics-based derivations of color distance metrics and semi-metrics that are shading and highlight invariant in RGB. In Chapter 4, various distance metrics and semi-metrics were reviewed. Each of those distance measures have drawbacks:

1. Euclidean distance is not an appropriate physics-based metric in RGB since it is highly intensity dependent. The same criticism applies to the Mahalanobis distance measure.

2. Vector angle, although shading invariant in RGB, is not an appropriate physics-based semi-metric since distance measures for low pixel values are unreliable.

As opposed to the Euclidean distance and the vector angle, statistical distance measures depend on stochastic information encoded in the data in addition to difference computation between the pixels being compared. In this chapter, we pose the following question: can a metric be designed to be shading and highlight invariant, as well as noise resistant?

There are several important issues and assumptions that need to be made under this basic premise:

1. A metric which is invariant to shading needs to factor out the illumination. We will assume that the illumination varies linearly with the pixel values (as the illumination intensity increases, so will the RGB pixel values increase). This assumption is not always valid especially for very low (dark regions) and very high (usually small highlight regions) intensity values. However, we are not dealing with perceptually correct color differences such as for example those achieved in CIE Luv or CIE Lab [122] where small differences might matter.

2. We will assume that the illumination of the color scene is white light. In [150] a simple linear transformation based on the highest intensity point in the image (which is assumed to have the characteristics of the illumination source) was used to white balance the image (*i.e.*, to ensure that the white light assumption is valid). However, this transformation is not necessarily easily done as a pixel with similar characteristics to the illumination color might not be found automatically. In all cases, images studied in this thesis were obtained (or are assumed to have been obtained) under white light and therefore we have no reason to believe that this assumption would not hold.

3. We will assume that we are working in the RGB or sensor space where we can easily apply the Dichromatic Reflection Model [123]. Therefore, we need not be concerned with transformations into a different color space.

4. Noise resistance can be achieved with respect to different types of noise. Noise can frequently introduce errors in measurement and it is important to take it into account. In this chapter, we consider white noise stemming from the image capture process which we model as an additive Gaussian distributed noise:

$$\underline{x} \;=\; \underline{a} + \underline{\upsilon}_x \tag{6.1}$$

   where $\underline{x}$ is the pixel vector, $\underline{a}$ is the true representation of $\underline{x}$, and $\underline{\upsilon}_x$ is Gaussian-distributed noise with covariance R_x that depends on $\underline{a}$. Therefore, the noise for each $\underline{x}$ is independent of the other noises.

   However, additive Gaussian noise is not necessarily a good assumption since real CCD camera noise is strongly dependent on the image intensity level and may vary

from color to color [63]. It includes mainly five noise sources: fixed pattern noise, dark current noise, shot noise, amplifier noise and quantization noise. Finding image regions in order to estimate the noise level for different intensities and colors is generally not possible in a single image since many regions will be too small [139]. We will therefore keep the Gaussian assumption though understand that it does not fully represent reality.

The chapter is organized in the following manner. First, we will demonstrate why the vector angle and Euclidean distance are not appropriate distance measures especially when we would like to measure intensity invariant distances in highly variable and dark areas. Next, we will introduce the hypothesis testing probabilistic framework which will be followed by the development of intensity invariant and noise resistant semi-metrics. Finally, we will discuss specular reflection or highlight invariance and modify our probabilistic semi-metrics in order to allow them to detect surfaces irrespective of specular reflections. We will show results throughout the chapter to illustrate our theoretical development.

6.1 Vector Angle Limitations

The distance between two pixels can be computed in several different ways in RGB. Using the Euclidean distance, the distance becomes intensity-dependent and, therefore, is not applicable to assessing differences only based on color. Figure 6.1 shows that there are high differences with respect to pixel intensity.

The vector angle distance measure (4.12) is effectively the square of the sine of the angle between two vectors. Figure 6.2 shows vector angle distances between various color pixels in RGB. For very low intensity pixel values in RGB, the distance measure behaves erratically. That is, for small changes in low intensity pixel values, the measured distance can be arbitrarily different. In other words, a small amount of noise will create vastly different results. This indicates that using the vector angle the statistics break down for very dim (or low intensity) pixels which invalidates its use even though the intensity invariant feature is very attractive.

We can also approximate the vector angle using the Euclidean distance on RGB points projected onto the unit sphere known as the normalized color space rgb [61]. This is shown

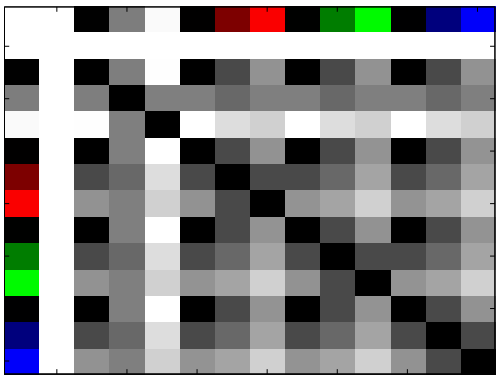
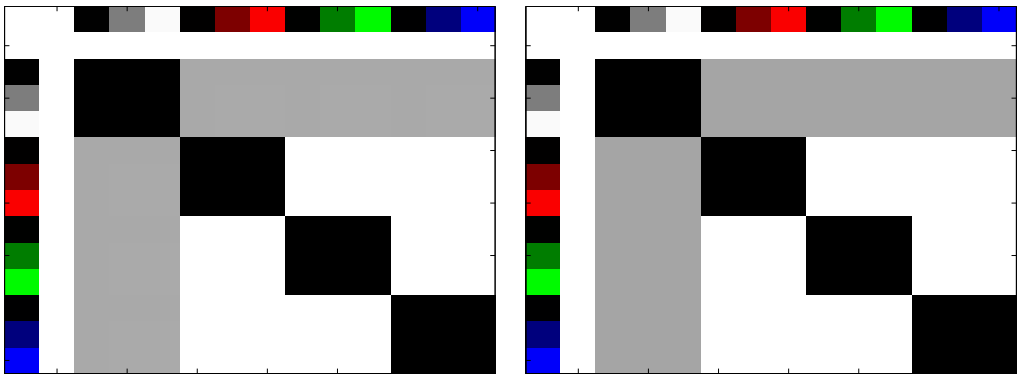


Figure 6.1: Euclidean distances between various colors in RGB are shown. A black square indicates that the colors are similar, while a white one shows high disagreement. Shades of gray illustrate the nuances in the color differencing results. There is a clear pattern of having low distances between pixels of similar intensity and large distances between pixels of very different intensities all without regard to the intrinsic pixel color. The RGB colors correspond to the following values (from left to right and from top to bottom): $\{1, 1, 1\}$, $\{125, 125, 125\}$, $\{250, 250, 250\}$, $\{1, 0, 0\}$, $\{125, 0, 0\}$, $\{250, 0, 0\}$, $\{0, 1, 0\}$, $\{0, 125, 0\}$, $\{0, 250, 0\}$, $\{0, 0, 1\}$, $\{0, 0, 125\}$, and $\{0, 0, 250\}$.



Vector angle in RGB Euclidean distance in rgb

Figure 6.2: Shading Invariant Distances Between Various Colors in vector angle in RGB and using Euclidean distance in the normalized space rgb. The color distances are identical and therefore we can use these formulations interchangeably. Notice that distances between pixels with low intensity values such as $\{1, 0, 0\}$ and $\{0, 1, 0\}$ show a very high degree of difference.

in Figure 6.3. Since $\sin\theta \approx \theta$ for similar colors, not much error is introduced. The color distances calculated using the Euclidean distance in normalized rgb are the same as the distances obtained using vector angle in RGB (see Figure 6.2).

6.2 Hypothesis Tests: Three Choices

In order to create a probability-based distance measure, we will first need to introduce the concept of hypothesis testing. Then we will describe two different hypothesis test-based distance measures that can be used as shading invariant methods with the desired characteristics.

There are three ways of formulating the problem:

1. The most commonly used method involves asking the question whether quantities (in this case pixel values) are from the same class. In essence, this corresponds to

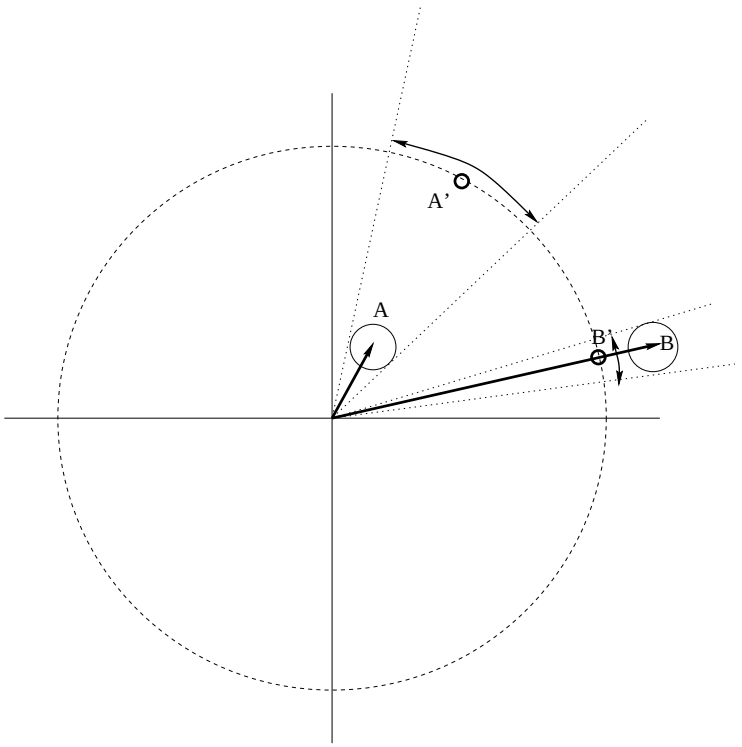


Figure 6.3: The normalized rgb color space is demonstrated through projecting RGB vectors onto the unit sphere (shown here in 2-D for ease of viewing). The RGB pixels A and B (thick black arrows) are projected onto the unit sphere at points A' and B' respectively (small circles on the unit circle). The dotted lines are indicative of the variance of the pixel values. Therefore, a pixel with low RGB intensities that is projected onto the unit sphere will have relatively higher variance on the unit sphere than pixels that have high RGB intensities.

the following hypothesis test:

$$H_0 : \underline{x} = \underline{y} \qquad (6.2)$$
$$H_1 : \underline{x} \neq \underline{y}$$

Expression H_0 is usually called the null hypothesis.

2. We could ask how similar a pixel is to a particular class or region prototype without regard to its relationship with adjacent pixels. This is the typical clustering problem such as k-means or mixture of Gaussians [35]. This can be characterized as

$$H_i : \underline{x} = \underline{a}_i \qquad (6.3)$$

where $i = 1, \ldots, K$ represents a class index, and $\{\underline{a}_i\}$ is the class mean. In this formulation, as the number of classes grows, the number of tests grows with it.

3. We could also ask to which class each pixel (of a set of adjacent pixels) belongs. The hypothesis test then becomes

$$H_{ij} : \quad \underline{x} = \underline{a}_i \quad \underline{y} = \underline{a}_j$$

This formulation is seldom used since as the number of classes grows, the number of tests grows quadratically quickly becoming unmanageable. However, we can ask some questions about $\underline{a}_i$ and $\underline{a}_j$ in order to reduce the number of tests to only one. For example, instead of having formal classes we can estimate $\underline{a}_i$ and $\underline{a}_j$ based on $\underline{x}$ and $\underline{y}$ by asking whether $\underline{x}$ and $\underline{y}$ are explained by the same mean (*i.e.*, $\underline{a}_i = \underline{a}_j$) or different means (*i.e.*, $\underline{a}_i \neq \underline{a}_j$).

In this chapter, probabilistic metrics are developed for the first and third options. The clustering formulation (the second option) is considered with a non-probabilistic distance measure in Chapter 7.

6.3 Same-Class Hypothesis Test

Noise can frequently introduce errors in measurement and it is important to take it into account. One way in which this can be done is to assume a particular noise model and

devise a dissimilarity measure based on this assumption. Consider two pixels $\underline{x}$ and $\underline{y}$ defined under model (6.1). The scalar case (*i.e.*, vectors of size 1) of the hypothesis test (6.3) becomes

$$x - y \sim \mathcal{N}(0, \sigma_x^2 + \sigma_y^2) \tag{6.4}$$

where σ_x^2 is the noise variance for pixel $\underline{x}$. We can state this since the variance of the sum of two independent random variables is the sum of their variances. Therefore, given pixel model (6.1), we assume that the noise distributions associated with $\underline{x}$ and $\underline{y}$ are independent. This leads to

$$\frac{x - y}{\sqrt{\sigma_x^2 + \sigma_y^2}} \sim \mathcal{N}(0, 1) \tag{6.5}$$

which can be restated as the probability

$$\Pr\left(\frac{|x - y|}{\sqrt{\sigma_x^2 + \sigma_y^2}} > c\right) = 2Q(c) \tag{6.6}$$

where c is some suitable value for which the hypothesis is true and Q corresponds to the cumulative normal distribution function:

$$Q(c) = \frac{1}{\sqrt{2\pi}} \int_c^\infty e^{\frac{-x^2}{2}} dx \tag{6.7}$$

This leads to the probabilistic metric based on the hypothesis test (6.3)

$$\Phi_Q(x, y) = -\ln\left[2Q\left(\frac{|x - y|}{\sqrt{\sigma_x^2 + \sigma_y^2}}\right)\right]. \tag{6.8}$$

It is easy to show that equation (6.8) satisfies the four metric conditions. However, we would like to obtain an approximation that is similar to other metric forms. To find an approximation for $\Phi_Q(x, y)$, we will first need to find an approximation for Q. Suppose we approximate Q as follows [42]

$$\begin{aligned} Q(c) &\approx \frac{1}{\sqrt{2\pi}c} \int_c^\infty x e^{\frac{-x^2}{2}} dx \\ &\approx \frac{1}{\sqrt{2\pi}c} e^{-\frac{c^2}{2}}. \end{aligned} \tag{6.9}$$

Taking the derivative of (6.9) with respect to c yields

$$\begin{aligned}\frac{\partial \Phi_Q}{\partial c} &= \frac{-1}{2Q(c)}\frac{\partial}{\partial c}(2Q(c)) \\ &= \frac{1}{2Q(c)} \cdot 2\frac{1}{\sqrt{2\pi}}e^{-\frac{c^2}{2}} \qquad (6.10) \\ &\approx \frac{e^{\frac{-c^2}{2}}}{\frac{1}{c}e^{\frac{-c^2}{2}}} = c. \qquad (6.11)\end{aligned}$$

Therefore, since the slope of $\Phi_Q(x, y)$ is approximately linear, $\Phi_Q(x, y)$ is approximately quadratic. Thus, we choose as our distance metric

$$\Phi_S(x, y) = c^2 = \frac{(x - y)^2}{\sigma_x^2 + \sigma_y^2} \qquad (6.12)$$

which is a quadratic function in terms of c. (6.12) is a generalization of the Mahalanobis distance [35] to a distance metric between quantities x and y which have different probability distribution functions and hence different variances σ_x^2 and σ_y^2 (the Mahalanobis distance assumes the same probability distribution function for all data points). Note, that we did not choose the Mahalanobis distance as a starting point; instead, a generalized Mahalanobis distance was derived from basic principles associated with the formulated problem. For different initial assumptions, a different metric would be derived.

We go back to the case of multidimensional vectors. In the multidimensional case, the hypothesis test becomes

$$\sqrt{(\underline{x} - \underline{y})^T(\underline{x} - \underline{y})} \sim \mathcal{N}(0, R_x + R_y) \qquad (6.13)$$

where R_x is the noise covariance matrix for RGB pixel $\underline{x}$. We cannot assume independence between color bands in RGB since they are correlated. However, we can sum the two covariance matrices R_x and R_y since they are independent. This leads to

$$\sqrt{(\underline{x} - \underline{y})^T(R_x + R_y)^{-1}(\underline{x} - \underline{y})} \sim \mathcal{N}(0, 1) \qquad (6.14)$$

which means that we are testing the magnitude of $\underline{x} - \underline{y}$ and similarly as in the scalar case leads to the metric

$$\Phi(x, y) = -\ln\left[2Q\left(\sqrt{(\underline{x} - \underline{y})^T(R_x + R_y)^{-1}(\underline{x} - \underline{y})}\right)\right] \qquad (6.15)$$

And therefore the corresponding approximation is

$$\Phi_S(\underline{x}, \underline{y}) = (\underline{x} - \underline{y})^T (R_x + R_y)^{-1} (\underline{x} - \underline{y}) \tag{6.16}$$

Thus the metric (6.16) is a generalization of the Euclidean distance between two points taking into account their individual noise statistics. This equation is of course only valid as long as the assumption of Gaussian statistics holds.

To make this semi-metric intensity invariant, we need to consider normalized color by projecting the RGB pixel vectors onto the unit sphere [61]. This means that distance measure (6.16) will become

$$\Phi_S(\bar{\underline{x}}, \bar{\underline{y}}) = (\bar{\underline{x}} - \bar{\underline{y}})^T (\bar{R}_x + \bar{R}_y)^{-1} (\bar{\underline{x}} - \bar{\underline{y}}) \tag{6.17}$$

where $\bar{\underline{x}} = \frac{\underline{x}}{|\underline{x}|}$ represents a normalized vector and $\bar{R}_x = \frac{R_x}{|\underline{x}|^2}$ is the normalized noise covariance matrix. In this formulation, $\bar{R}_x$ now varies with the magnitude of $\underline{x}$. Because of this dependence (6.17) is a semi-metric as the triangle inequality is no longer satisfied which was still the case for (6.16).

Figure 6.4 shows the distance computations using the same-class hypothesis (SCH) distance (we will call this distance the same-class hypothesis distance measure since the null hypothesis tests whether the two pixel vectors are the same, *i.e.*, in the same class). The distances between dark pixels are now a bit lower than distances between dark pixels and other colors with higher intensities (*e.g.*, compare distances between "dark red" and "dark green" and between "medium red" and "dark green" in Figures 6.2 and 6.4). Furthermore, the distances between colors with high intensity values are large as they were before. This analysis suggests that low intensity pixels will most likely merge with pixels of high intensity. This is desirable in areas where a shadow falls upon an object which results in some parts of the object being very dark. However, other areas which are adjacent to this dark region might become merged with it.

6.4 Common Mean Hypothesis Test

An original approach using the Common Mean Hypothesis (CMH) test (6.4) involves the estimation of $\underline{a}_i$ based on the values of $\underline{x}$ and $\underline{y}$ in order to find the $\underline{a}_i$ that maximizes the

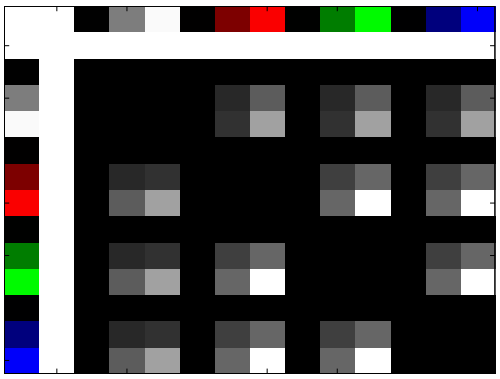


Figure 6.4: SCH shading invariant distances between various colors in RGB using equation (6.17). Notice that distances between pixels with low intensity values are now very small compared to distances between higher intensity pixels while remaining "0" for pixels of exactly the same color.

joint probability distribution $p(\underline{x}, \underline{y}|\underline{a}_i)$. We can also look for the $\underline{a}_i$ that is equidistant in terms of standard deviations from $\underline{x}$ and $\underline{y}$, in other words the $\underline{a}_i$ for which $p(\underline{x}|\underline{a}_i) = p(\underline{y}|\underline{a}_i)$ is true. For simplicity of notation and without loss of generality, we will use $\underline{a} = \underline{a}_i$ for the developments in this section and the next one.

If we first assume that the means $\underline{a}$ are known, then we can estimate the likelihood $p(\underline{x}|\underline{a})$ using

$$p(\underline{x} \mid \underline{a}) = \frac{1}{\sqrt{2\pi} \mid R_x \mid} e^{-\frac{1}{2}(\underline{x}-\underline{a})^T R_x^{-1}(\underline{x}-\underline{a})}. \tag{6.18}$$

(6.18) gives us the well-known measure of how likely it is for $\underline{x}$ to come from a distribution with mean $\underline{a}$. Note that we can transform this probability into a distance measure by using $-\ln p(\underline{x}|\underline{a})$. We now assume that the prior means $\underline{a}$ are unknown and ask the question what is the likelihood that there is a common $\underline{a}$ which explains both $\underline{x}$ and $\underline{y}$?

Through the independence property (we can assume that $\underline{x}$ and $\underline{y}$ are conditionally independent since their noises are independent), we can state that $p(\underline{x}, \underline{y} \mid \underline{a}) = p(\underline{x} \mid \underline{a})p(\underline{y} \mid \underline{a})$. Therefore, given that we would like to find out how consistent $\underline{x}$ and $\underline{y}$ are with respect to each other, the desired likelihood is

$$p(\underline{x}, \underline{y}) = \max_{\underline{a}} p(\underline{x}, \underline{y} \mid \underline{a}) \tag{6.19}$$

where the max operator is computed over all possible $\underline{a}$. This can be a computationally expensive task using an exhaustive search, as well as depending on whether the set of reals or integers is used to represent $\underline{a}$. Making the conditional independence assumption we have

$$p(\underline{x}, \underline{y}) = p(\underline{x} \mid \underline{a})p(\underline{y} \mid \underline{a}) \tag{6.20}$$

which can also be written by

$$\Phi(\underline{x}, \underline{y}) = -\ln[p(\underline{x} \mid \underline{a})] - \ln[p(\underline{y} \mid \underline{a})]. \tag{6.21}$$

The distance metric can, therefore, be represented by

$$\Phi_C(\underline{x}, \underline{y}) = (\underline{x} - \underline{a})^T R_x^{-1}(\underline{x} - \underline{a}) + (\underline{y} - \underline{a})^T R_y^{-1}(\underline{y} - \underline{a}). \tag{6.22}$$

This distance measure is very easy to calculate and has no trouble with dark pixels assuming that we can find a good $\underline{a}$ by, for example, minimizing (6.22) with respect to $\underline{a}$. However, (6.22) is dependent on intensity and will not work with illumination-dependent error covariances. To transform (6.22) into an intensity invariant measure we follow the same methodology as in the previous section. Namely, we project the vectors $\underline{x}$, $\underline{y}$ and $\underline{a}$ onto the unit sphere. Then, we obtain the following distance measure:

$$\Phi_C(\bar{\underline{x}}, \bar{\underline{y}}) = (\bar{\underline{x}} - \bar{\underline{a}})^T R_x^{-1}(\bar{\underline{x}} - \bar{\underline{a}}) + (\bar{\underline{y}} - \bar{\underline{a}})^T R_y^{-1}(\bar{\underline{y}} - \bar{\underline{a}}). \tag{6.23}$$

This distance measure is very easy to calculate, has no trouble with dark pixels, is intensity invariant and will work with non-trivial illumination-dependent error covariances.

$\bar{\underline{a}}$ is not considered to be a region prototype that is commonly accepted in the literature [35]. Rather, the idea is to find an $\bar{\underline{a}}$ which best explains both $\bar{\underline{x}}$ and $\bar{\underline{y}}$ and to use this intermediary quantity as a means of assessing the distance between $\bar{\underline{x}}$ and $\bar{\underline{y}}$. As mentioned at the beginning of this section, there are two different ways to determine $\bar{\underline{a}}$. First, we could determine the $\bar{\underline{a}}$ which maximizes the joint conditional probability $p(\bar{\underline{x}}, \bar{\underline{y}}|\bar{\underline{a}})$. Second, we could find the $\bar{\underline{a}}$ which best fits $p(\bar{\underline{x}}|\bar{\underline{a}}) = p(\bar{\underline{y}}|\bar{\underline{a}})$.

6.4.1 Finding the minimum mean

To find the optimum $\bar{\underline{a}}$ on the unit sphere we minimize (6.23) with respect to $\bar{\underline{a}}$. To keep the analysis tractable we will assume that $R_x = \sigma_x^2 I$ and $R_y = \sigma_y^2 I$ and for the intensity

invariant case we have $\bar{R}_x = \bar{\sigma}_x^2 I$ and $\bar{R}_y = \bar{\sigma}_y^2 I$. In order to minimize (6.23), we perform a component-based differentiation with respect to $\bar{\underline{a}}_j$ where j represents the jth component of the vector, and set these partial derivatives to 0, *i.e.*, $\frac{\partial}{\partial \bar{a}_j} p(\bar{\underline{x}}, \bar{\underline{y}} | \bar{\underline{a}}_j) = 0$. For each component $\bar{a}_j$ the we have

$$-2\bar{\sigma}_x^{-2}(\bar{x}_j - \bar{a}_j) - 2\bar{\sigma}_y^{-2}(\bar{y}_j - \bar{a}_j) = 0. \tag{6.24}$$

Rearranging terms we obtain

$$\bar{a}_j = \frac{\frac{\bar{x}_j}{\bar{\sigma}_x^2} + \frac{\bar{y}_j}{\bar{\sigma}_y^2}}{\bar{\sigma}_x^{-2} + \bar{\sigma}_y^{-2}} \tag{6.25}$$

which simplifies to

$$\bar{a}_j = \frac{\bar{x}_j \bar{\sigma}_y^2 + \bar{y}_j \bar{\sigma}_x^2}{\bar{\sigma}_x^2 + \bar{\sigma}_y^2} \tag{6.26}$$

or in vector notation

$$\bar{\underline{a}} = (\bar{\underline{x}}^T \bar{R}_y + \bar{\underline{y}}^T \bar{R}_x)(\bar{R}_x + \bar{R}_y)^{-1}. \tag{6.27}$$

Note that if $|\bar{\underline{x}}| = 0$ or $|\bar{\underline{y}}| = 0$, $\bar{\underline{a}}$ will correspond to the zero vector which is the desired behavior.

6.4.2 Finding an equally likely mean

Another way to compute $\bar{\underline{a}}$ is to find the mean which equally accommodates $\bar{\underline{x}}$ and $\bar{\underline{y}}$. This results in the following formulation

$$(\bar{\underline{x}} - \bar{\underline{a}})^T \bar{R}_x^{-1} (\bar{\underline{x}} - \bar{\underline{a}}) = (\bar{\underline{y}} - \bar{\underline{a}})^T \bar{R}_y^{-1} (\bar{\underline{y}} - \bar{\underline{a}}). \tag{6.28}$$

Therefore, on a component basis we have

$$\frac{(\bar{x}_j - \bar{a}_j)^2}{\bar{\sigma}_x^2} = \frac{(\bar{y}_j - \bar{a}_j)^2}{\bar{\sigma}_y^2}. \tag{6.29}$$

The solution to this equation produces an $\bar{\underline{x}}$ that is equidistant from $\bar{\underline{x}}$ and $\bar{\underline{y}}$ in units of standard deviation $\bar{\sigma}_x$ and $\bar{\sigma}_y$. If we apply the square root operator to both sides of (6.29) and rearrange terms, we obtain:

$$\bar{a}_j = \frac{\bar{x}_j \bar{\sigma}_y + \bar{y}_j \bar{\sigma}_x}{\bar{\sigma}_x + \bar{\sigma}_y}. \tag{6.30}$$

Note that the variables $\underline{\bar{x}}$ and $\underline{\bar{y}}$ are multiplied through each other's standard deviation and not variance as was the case for (6.26).

6.4.3 Discussion

Both (6.27) and (6.30) offer plausible choices for $\underline{\bar{a}}$. Figures 6.5 and 6.6 show the distance matrix as before for (6.23) using (6.27) and (6.30) respectively. Notice that Figures 6.5 and 6.4 are identical. In fact, it is easy to show that substituting (6.27) into (6.23) will yield (6.17). This indicates that (6.17) is a simplification of (6.23) when means are obtained using (6.27). Therefore, we no longer need to compute means $\underline{\bar{a}}$ for the CMH distance measure through function minimization reducing somewhat the computational complexity of each site visit. Certainly the same conclusions apply to (6.23) with $\underline{\bar{a}}$ represented using (6.27) as did to (6.17).

The results for Figure 6.6 are different from Figure 6.5. Whereas differences between achromatic colors such as the color white (third color) and low intensity chromatic colors such as the very low intensity red (color 4) were small in Figure 6.5, they are now large. This implies that pixels with various shades of gray will not automatically be merged with adjacent regions which was the case with (6.17). Furthermore, differences between low intensity chromatic pixels are now more nuanced. Whereas perviously a factor of four or five separated them, now they are separated by a several orders of magnitude which gives us more freedom for setting image segmentation parameters.

6.5 Preliminary Results

We will perform the segmentation using the SNA-GM-ICM with the edge Potts model with different β schedules. The number of labels used will be $K = 10$ and relabelling will occur after every level. We will assume that the variance of the noise σ_i^2 for each color band $i \in \{R, G, B\}$ is the same and is given by the user. For the peppers image we assume a $\sigma_i^2 = 8^2$ while for the toy[1] image a $\sigma_i^2 = 4^2$ was chosen.

[1] "toy" image [50] publicly available at the following website
`http://www.science.uva.nl/research/isla/themes/FeaturesAndColor.php`.

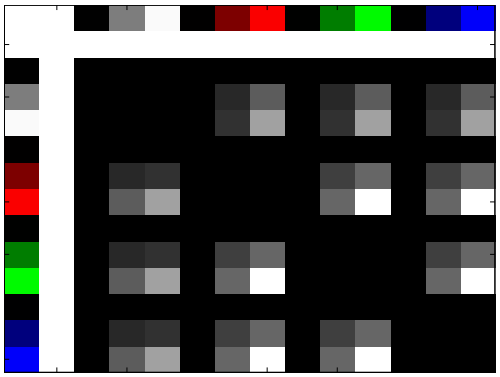


Figure 6.5: Distances between various colors are indicated for the Common Mean Hypothesis (CMH) (6.23) with a mean (6.27). This figure is identical to Figure 6.4.

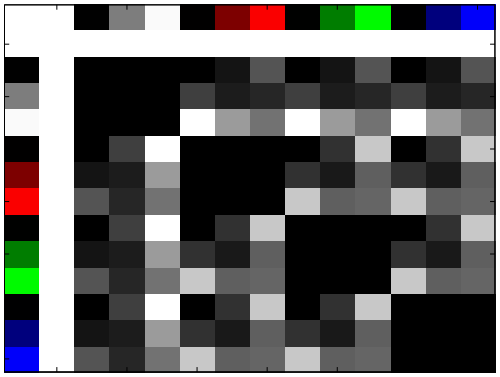


Figure 6.6: Distances between various colors are indicated for the Common Mean Hypothesis (CMH) (6.23) with a mean (6.30). Notice that distances between pixels with low intensity values are now very small compared to distances between higher intensity pixels while remaining "0" for pixels of exactly the same color.

Figure 6.7 shows an example of the segmentation of the peppers image. As opposed to the vector angle results where areas of dark pixels contained a proliferation of small regions, the results for the SCH distance measure show that pixels in dark regions can be grouped together or grouped with an adjoining region with a well defined color. However, regions of different color can still merge together if they are connected by a darker region. This is a highly undesirable side-effect of the devised semi-metric. Also, note that highlight areas are detected as separate regions as previously in Figures 5.24 and 5.32.

Figure 6.8 shows the results for the toy image. Note that vector angle results for a lower β are generally worse especially around highlight areas whereas for a higher β the resulting image appears much cleaner; however, one object (in the upper left panel) has been completely absorbed by the background from the image. Regions of very low intensity help to separate objects from the background (*e.g.*, the ball in the lower left panel). For the SCH color distance, regions of constant color are grouped together and separated from others. One very dark shadow in the lower left panel causes the ball object to merge with the background color (expected result). Many edge pixels are grouped separately from the objects due to jpeg compression artifacts in the image. Furthermore, the proliferation of small regions along the edges causes large regions such as the background and the sphere to have only a few pixels of common border. If the distance across those pixels is small then region merging will occur and edge regions will be left unmerged. This error would be preventable by using a mean model with the SCH color distance. The pixels which is used to compute the mean would be weighted by the corresponding variances (*i.e.*, low intensity pixels would be weighed less than higher intensity ones). This is left as a future exercise.

Figure 6.9 shows results for the common mean hypothesis distance measure with equally likely mean. Not surprisingly, the results are similar to those in Figure 6.7 with perhaps slightly better performance shown by less region-to-region spilling. As before, highlights appear as separate segmented objects. Both of these distance measures with SNA-GM-ICM took on average 230 seconds, 22 levels and 1.8 million site visits. For the toy image using SNA-GM-ICM the statistics were 60 seconds running time, 330000 site visits and 25 nested levels.

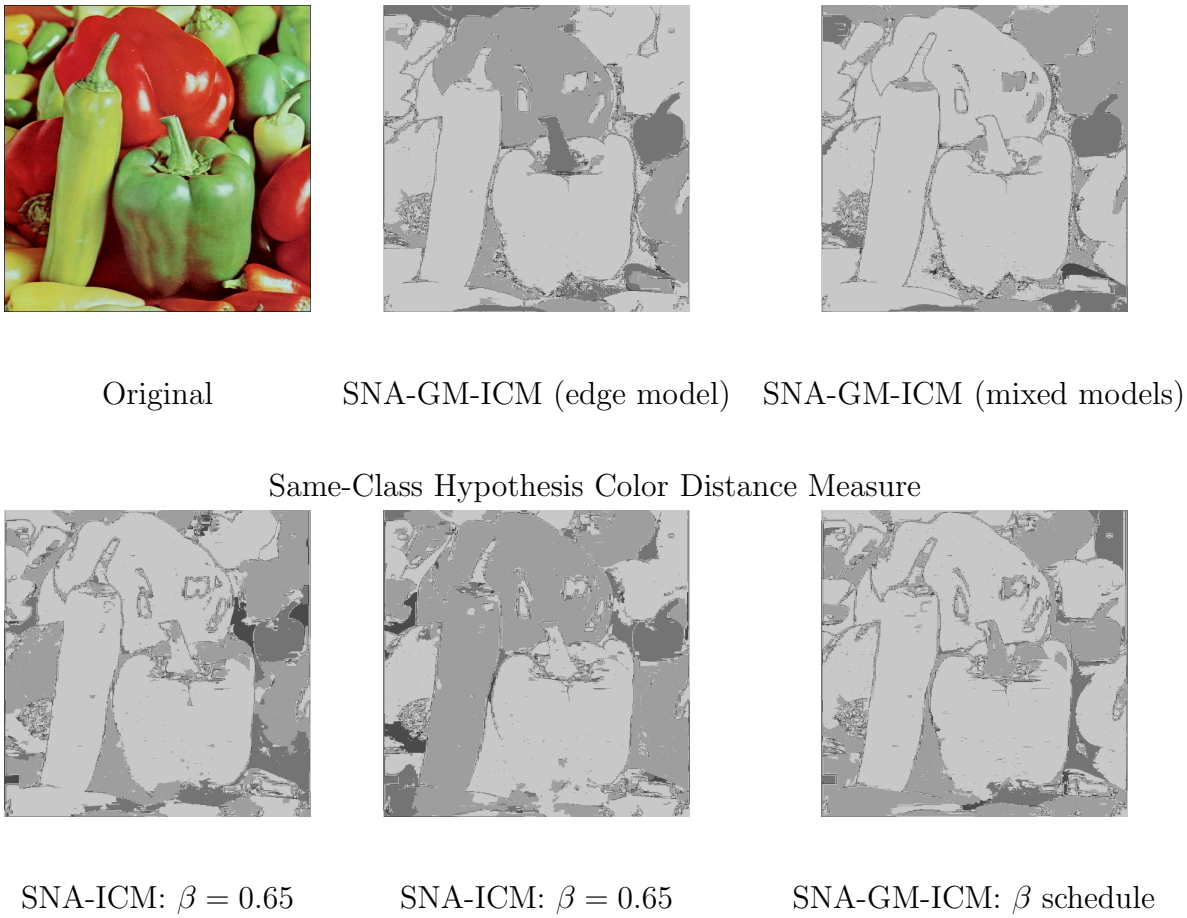


Figure 6.7: Results on the peppers image using the Same-Class Hypothesis (SCH) color distance measure (6.17) in RGB. SNA-GM-ICM used with vector angle with edge model ($\beta = \{0.0010, 0.0020, 0.0030, 0.0040, 0.0045\}$) and with mixed models ($\beta = \{0.0010, 0.0011, 0.0012, \cdots, 0.0044, 0.0045\}$). SNA-GM-ICM used with SCH had a schedule of $\beta = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.65\}$. Note that dark regions appear as individual regions or are merged with similarly colored lighter regions. Some regions of differing color are merged together usually through the intermediary of a dark region.

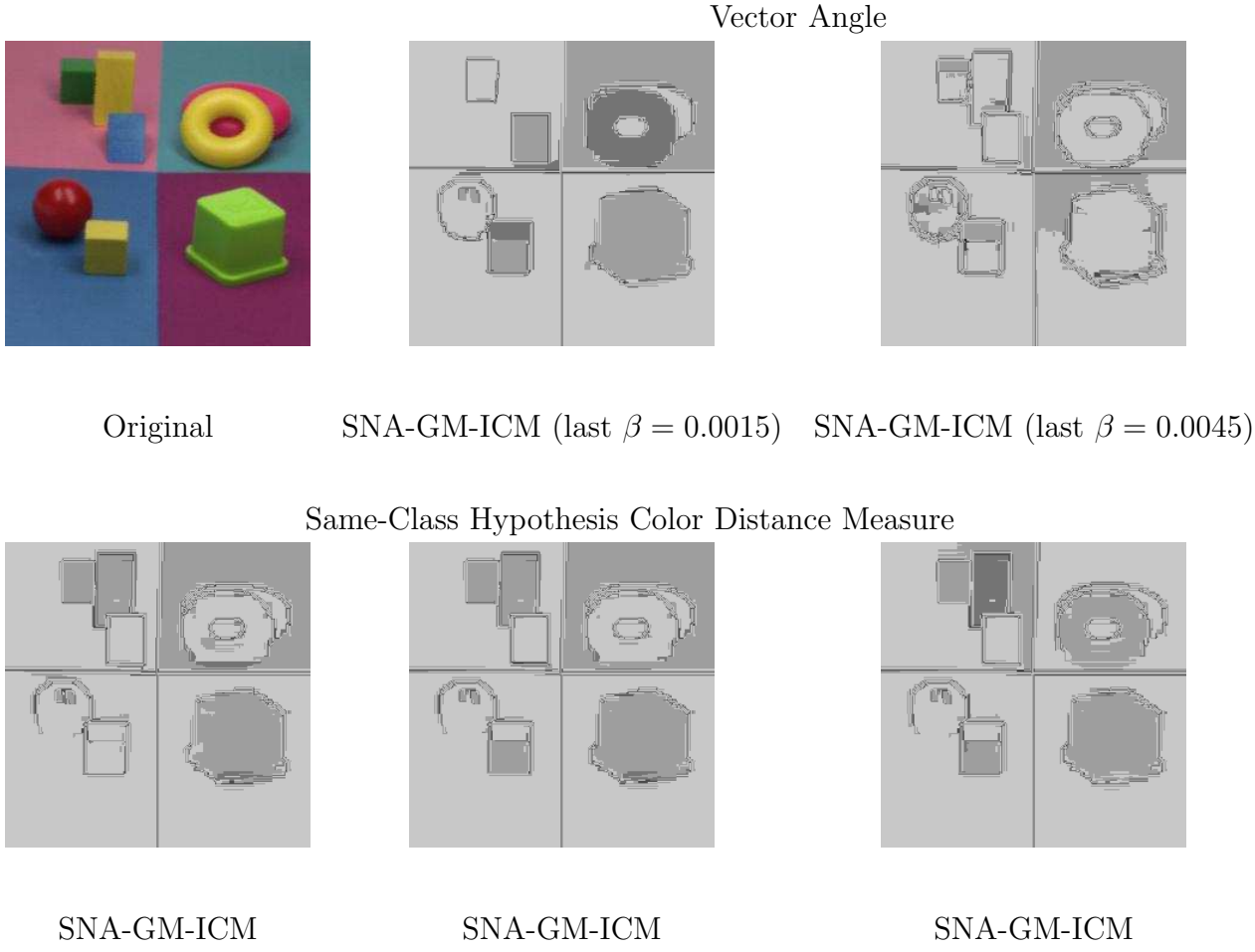


Figure 6.8: Results on the toys image (size: 256×256) using the Same-Class Hypothesis color distance measure (6.17) in RGB. The β schedule for the first SNA-GM-ICM with vector angle was $\beta = \{0.0010, 0.0020, 0.0030, 0.0040, 0.0045\}$ and for the second it was $\beta = \{0.0005, 0.001, 0.0015\}$). A higher β is needed to produce less noisy vector angle-based images at the cost of one region merging into the background from the result. SNA-GM-ICM used with SCH results were obtained using a schedule of $\beta = \{0.25, 0.5, \ldots, 2.25, 2.5\}$. Some regions of differing color are merged together usually through the intermediary of a dark region. Edge areas produce many small regions due to jpeg compression artifacts in the image.

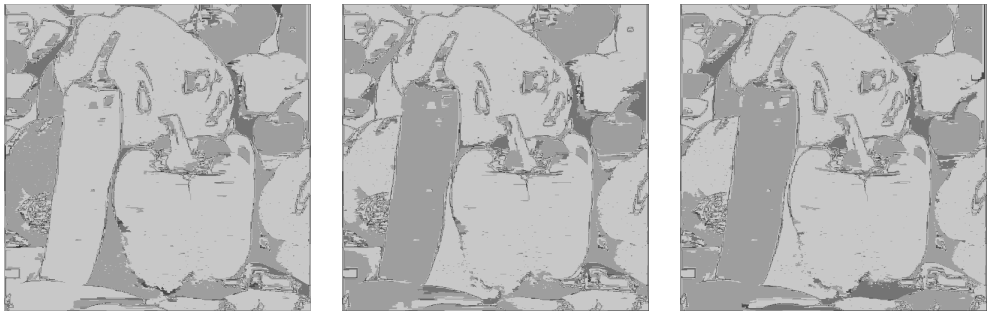


Figure 6.9: Results on the peppers image using the Common Mean Hypothesis distance measure (6.23) with equally likely mean (6.30). Examples of results for SNA-GM-ICM with a schedule of $\beta = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.65\}$. Note that dark regions appear as individual regions or are merged with similarly colored lighter regions. Some regions of differing color are merged together usually through the intermediary of a dark region. Three separate results are shown.

6.6 Color Spaces: Highlight Invariance Projections

Since shading invariance has been achieved using probabilistic semi-metrics, we now turn our attention to specular reflection which is quite problematic for image segmentation (cf. Section 4.2). Specular reflection or highlight invariance can be achieved for example by using the vector projection method of Tominaga [133] where the average intensity of the pixel is subtracted from each pixel component such that:

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix} - \begin{bmatrix} R+G+B \\ R+G+B \\ R+G+B \end{bmatrix} /3. \tag{6.31}$$

In this transformation, the reflectance variation caused by specular reflection is removed by projecting the observed reflectance in an n-dimensional vector space along the illumination vector onto an (n-1)-dimensional subspace that is perpendicular to the illumination vector [133]. Therefore from a 3-dimensional RGB space we now have a 2- dimensional highlight invariant space where one of $\{R', G', B'\}$ is a linear combination of the other two bands.

From a practical point of view, a 3-D plot of the RGB pixels values making up a

Figure 6.10: Fruits image.

homogeneously-colored region containing a highlight patch would show two connected clusters (one for the homogenous color and one for the highlight). For example, given the original image in Figure 6.10, Figure 6.11 shows a distribution of RGB pixels for four differently colored fruits. The four clusters (obtained using the Mixture of Principal Components algorithm [150]) appear highly spread-out and are non-linear (do not lie along a straight line in RGB space), because each cluster is composed of both body and specular reflections.

(6.31) transforms each set of nonlinear clusters into a single linear cluster representing the body reflection. This is well illustrated in Figure 6.12, where the original nonlinear clusters now appear as linear groupings. Given that the RGB components are assumed to be white balanced, the application of (4.5) and (6.31) eliminates the interface reflection term and reduces the sensor responses to [2]

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \alpha(x) \int S^o(\lambda, x) E(\lambda) \frac{1}{3} \begin{bmatrix} 2\mathcal{R}_R(\lambda) - \mathcal{R}_G(\lambda) - \mathcal{R}_B(\lambda) \\ -\mathcal{R}_R(\lambda) + 2\mathcal{R}_G(\lambda) - \mathcal{R}_B(\lambda) \\ -\mathcal{R}_R(\lambda) - \mathcal{R}_G(\lambda) + 2\mathcal{R}_B(\lambda) \end{bmatrix} d\lambda \quad (6.32)$$

$$= \alpha(x) \int S^o(\lambda, x) E(\lambda) \begin{bmatrix} \mathcal{R}'_R(\lambda) \\ \mathcal{R}'_G(\lambda) \\ \mathcal{R}'_B(\lambda) \end{bmatrix} d\lambda \quad (6.33)$$

This formulation is dependent on the shading factor (illumination) and the body reflection (material color), which makes this color representation highlight invariant. Individual

[2] A preliminary version of this work is archived in [150].

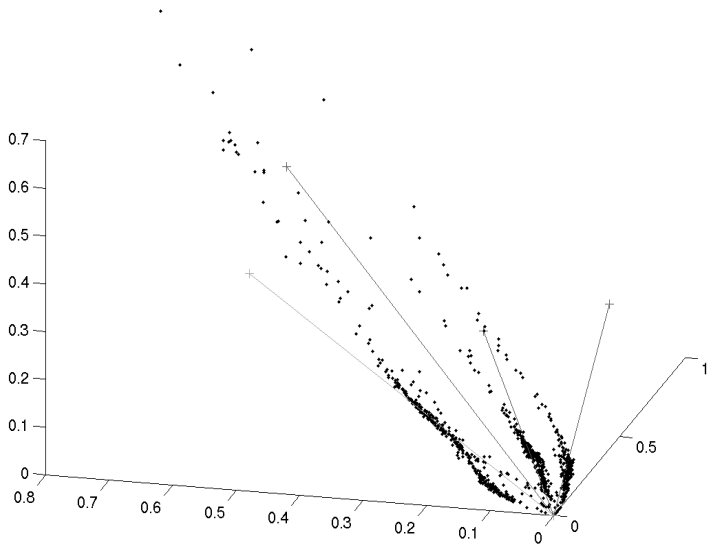


Figure 6.11: Distribution of pixels in the *RGB* space from the original image in Figure 6.10. The straight lines are the principal vectors obtained with the best mean squared error fit using the Mixture of Principal Components [150]. Both the red and orange fruits have been clumped into one larger cluster. Whereas three of the four clusters depicted in the image correspond to fruit colors, the fourth represents all of the highlight areas.

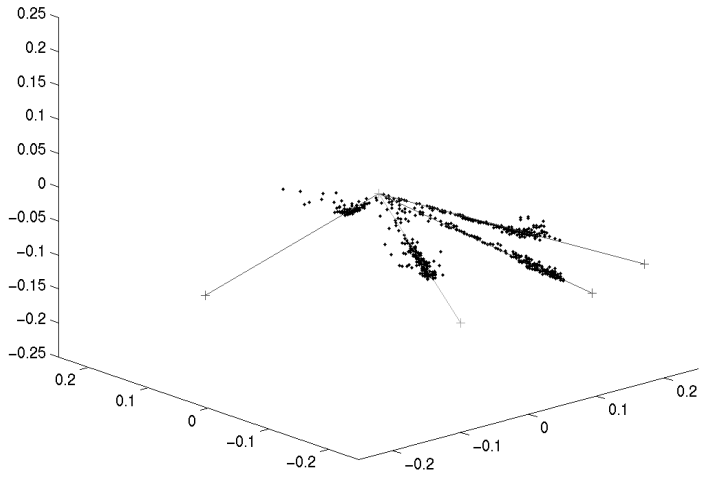


Figure 6.12: Distribution of pixels in the $R'G'B'$ space from Figure 6.10(a). Compare with the RGB distribution in Figure 6.11. The straight lines are the principal vectors obtained with the best MSE fit from [150]. The alignment of the four cluster prototypes with the four color clusters is clearly seen; each of the four cluster prototypes corresponds to a colored fruit.

elements of the pixel vector in the new representation will be shifted according to the average of the body reflection term. This results in the new space having negative coordinates. Equivalently the spectral sensitivity functions, $\mathcal{R}'_R(\lambda)$, $\mathcal{R}'_G(\lambda)$, and $\mathcal{R}'_B(\lambda)$, in the new system also have negative values.

Three properties were derived from this representation [150]. The first property states that all *RGB* colors fall into one of six quadrants in the new space. The second one indicates that all gray values (including saturated highlight areas) naturally collapse to the $(0, 0, 0)$ point. Pixels that are nearly saturated in intensity and nearly gray in color will be projected near to $(0, 0, 0)$. Finally, the third property demonstrates that the same color can only exist in quadrants that have at least one adjacent edge.

Highlight invariance can also be reformulated into a 2-dimensional representation as indicated earlier. Consider that (6.31) can be rewritten as

$$\begin{aligned} R' &= \frac{2R - G - B}{3} \\ G' &= \frac{2G - R - B}{3} \\ B' &= -R' - G' \end{aligned} \tag{6.34}$$

This can be summarized in the following linear transformation

$$\begin{bmatrix} R' \\ G' \end{bmatrix} = H \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{6.35}$$

where

$$H = \begin{bmatrix} 2/3 & -1/3 & -1/3 \\ -1/3 & 2/3 & -1/3 \end{bmatrix} \tag{6.36}$$

is the highlight invariant transformation matrix. We will use this transformation to modify (6.17) and (6.23) into highlight invariant color distance semi-metrics. First, however, we address the issue of modifying vector angle to adapt it to a highlight invariant space.

6.7 Vector Angle for Highlight Invariance

In order to be able to use the vector angle distance measure (4.12) in a space with negative coordinates we need to ensure that we can correctly determine the distance between vectors since we no longer operate in only the positive quadrant. The cosine of an angle (*i.e.*, the traditional definition of vector angle) denoted by $\cos\theta$ produces values in the range $[-1, 1]$. However, $\cos^2\theta$ only produces values in the compressed range $[0, 1]$ due to the squaring operation. Computing the distance measure (4.12) in standard RGB or normalized rgb was not a problem since all vectors had positive values. However, since the highlight invariant transformation introduces negative coordinates, we have $\frac{\underline{x}_i{}^T\underline{x}_j}{|\underline{x}_i|\cdot|\underline{x}_j|} \neq \left|\frac{\underline{x}_i{}^T\underline{x}_j}{|\underline{x}_i|\cdot|\underline{x}_j|}\right|$.

In order not to compute the arcsin to obtain the angle in (4.12), we will force the multiplier of $\left(\frac{\underline{x}_i{}^T\underline{x}_j}{|\underline{x}_i|\cdot|\underline{x}_j|}\right)^2$ to carry the sign of $\frac{\underline{x}_i{}^T\underline{x}_j}{|\underline{x}_i|\cdot|\underline{x}_j|}$. This leads to the equation:

$$\Phi_{VH}(i,j) = 1 - \frac{\underline{x}_i{}^T\underline{x}_j}{|\underline{x}_i|\cdot|\underline{x}_j|}\left|\frac{\underline{x}_i{}^T\underline{x}_j}{|\underline{x}_i|\cdot|\underline{x}_j|}\right| \tag{6.37}$$

which gives values in the range of $[0, 2]$; *i.e.*, for all values where $\frac{\underline{x}_i{}^T\underline{x}_j}{|\underline{x}_i|\cdot|\underline{x}_j|}$ is negative the range of distances is $(1, 2]$. This is a semi-metric as it does not satisfy the triangle inequality condition. Note that many pixels now map to the null set where vector angle is undefined. For this reason, a prototype-based Markov Random Field model was implemented to cope with the vector angle uncertainty. This research is described in Chapter 7. In the next section, we will use (6.37) with highlight invariance transformation (6.36).

6.8 Probabilistic Highlight and Shading Invariant Distance Measures

To make a highlight invariant distance measure, we will first transform the RGB space into the reduced $R'G'$ space and then carry out a similar analysis to the shading invariant distance measures. We will first analyze the transformation of the SCH distance measure (6.17) since it does not require the computation of a mean followed by the CMH distance measure (6.23) with equally likely mean (6.30).

We can easily transform our pixel definition (6.1) under the transformation (6.35). This will yield the following new pixel model:

$$\begin{aligned} \tilde{\underline{x}} &= H(\underline{a} + \underline{v}_x) \\ &= H\underline{a} + H\underline{v}_x \\ &= \tilde{\underline{a}} + \tilde{\underline{v}}_x \end{aligned} \tag{6.38}$$

where $\tilde{\underline{v}}_x$ is the noise component with covariance

$$\begin{aligned} \tilde{R}_x &= HR_xH^T \\ &= \sigma_x^2 \begin{bmatrix} 2/3 & -1/3 \\ -1/3 & 2/3 \end{bmatrix} \\ &= \sigma_x^2 H' \end{aligned} \tag{6.39}$$

Even if we assumed a diagonal covariance R_x, this assumption no longer holds for $\tilde{R}_x$. With these preliminaries we are now ready to transform the probabilistic color distance measures.

6.8.1 Same-Class Hypothesis Test

Distance measure (6.17) can be reformulated based on the new pixel model (6.38). Since we are projecting onto the unit circle to obtain intensity invariance our new quantities will be a new normalized color vector $\bar{\underline{x}} = \frac{\tilde{\underline{x}}}{|\tilde{\underline{x}}|}$ and a new normalized noise covariance matrix $\bar{\tilde{R}}_x = \frac{\tilde{R}_x}{|\tilde{\underline{x}}|^2}$. Thus, the new highlight invariant and shading or intensity invariant distance measure is

$$\Phi(\bar{\underline{x}}, \bar{\underline{y}}) = (\bar{\underline{x}} - \bar{\underline{y}})^T (\bar{\tilde{R}}_x + \bar{\tilde{R}}_y)^{-1} (\bar{\underline{x}} - \bar{\underline{y}}) \tag{6.40}$$

In this formulation, $\bar{\tilde{R}}_x$ now varies with position on the unit circle since the color bands R' and G' of the new highlight invariant space are correlated. Previously the noise covariance matrix was dependent on the magnitude of the color vector. Also note that the inverse of H' is

$$H'^{-1} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{6.41}$$

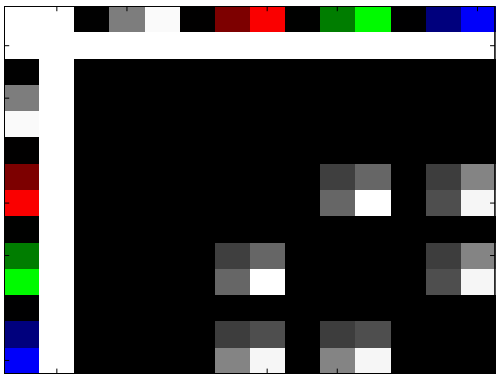


Figure 6.13: Highlight Invariant Same Class Hypothesis (HI-SCH) distances (6.40) between various colors are indicated. Notice that distances between pixels with gray level values (including very dark and very bright) and all other pixels are now "0" just like the differences for pixels of exactly the same color. This computation assumes that $\bar{\bar{R}}_x = \infty$ and $\bar{\bar{R}}_y = \infty$ for the zero vector. Of course in practice, those would be very large numbers and therefore the distances would be negligibly above zero.

which simplifies the computation of the matrix inverse and makes the distance measure easier to implement.

Figure 6.13 shows the distance computations using the highlight invariant same-class hypothesis (HI-SCH) distance. The distances between achromatic and chromatic pixels are now all 0. This means that all pixels which are very bright such as highlights should merge with the surrounding regions. At the same time pixels which are dark will also merge with surrounding color pixels. One undesirable side-effect is that gray values that are neither light or dark will also be merged in with surrounding chromatic pixels. This is inevitable since all achromatic pixels map to $\{0, 0, 0\}$ under the highlight invariant transformation (6.35).

Furthermore, the distances between colors with high intensity values are large as they were before. This analysis suggests that achromatic, as well as low intensity pixels will most likely merge with chromatic pixels of medium-to-high intensity. Areas that are naturally achromatic (*e.g.*, gray roads, white objects, etc.) will merge together with the neighboring chromatic regions or objects.

6.8.2 Common Mean Hypothesis Test

Making the Common Mean Hypothesis Test-based distance measures highlight invariant is a much more difficult task since the derivation of the mean $\underline{\bar{\bar{a}}}$ based on pixels $\underline{\bar{\bar{x}}}$ and $\underline{\bar{\bar{y}}}$ is dependent on the covariance matrix. Since the covariance is no longer diagonal in nature there is a dependence between the two vector components $\bar{\bar{a}}_i$ (where $i = 1, 2$) which significantly complicates the distance computation process since it requires using a gradient descent method to converge onto plausible choices for $\underline{\bar{\bar{a}}}$ given each combination of $\underline{\bar{\bar{x}}}$ and $\underline{\bar{\bar{y}}}$. Given that the concept of highlight and shading invariance is successfully demonstrated based on the Same Class Hypothesis test, making the Common Mean Hypothesis Test Distance Measure highlight invariant is left as future work.

6.9 Results

The distance measure (6.40) was used with SNA-GM-ICM and SNA-ICM. The standard deviation for the zero vector was set to $\sigma = 100$ which is large compared with the standard deviation for all other vectors. Results are shown on the peppers image in Figure 6.14. It is clear that highlights and shaded areas (such as the dark regions between the peppers) have been integrated into the surrounding regions. Note that the pepper objects appear mostly noise free while there is a proliferation of small regions at the edges between objects of different color. Furthermore, SNA-ICM using vector angle (with highlight invariance) results show that where dark regions and highlights appear, there is a proliferation of small regions.

In Figure 6.15, results for the toy image are shown. Vector angle together with the highlight invariance transformation achieves excellent segmentation results due to the noisy nature of distances in very dark areas which keep different regions apart (whereas in the case of the probabilistic distance measure, chromatic regions could be merged together because they both border a very dark intensity area). Also, note that the highlights in the toys image are not saturated which makes the image much easier to segment for vector angle. In the case of the highlight invariant SCH distance measure, shading and highlight invariance is achieved. There is some region-to-region spilling through dark areas due to the characteristics of the semi-metric.

In Figure 6.16, the saturated highlight on the Pooh image[3] is subsumed into the regions in the case of the probabilistic metric but not in the case of vector angle. However, the first vector angle-based segmentation looks much better than all the others. A noise standard deviation of $\sigma = 4$ was used for this image. The region merging in the case of the probabilistic distance measure might have been avoided with the use of a mean-based method to compute region-to-region distances on higher levels of the hierarchy. This should be part of a future development.

6.10 Summary

In this chapter, we have shown three intensity invariant and noise resistant distance measures and modified one of those measures to be also a highlight invariant distance measure. We have also introduced a new vector angle measure for a highlight invariant space. The distance measures have been derived from first principles and are well grounded in statistics. All shading invariant distance measures are semi-metrics as the triangle inequality is no longer satisfied. These semi-metrics allow the extension of vector angle (and the Euclidean distance in rgb) to highlight invariance and especially noise resistance. The effectiveness of these semi-metrics has been shown on image several segmentation results. However, some segmentation results suffer from region-to-region spilling which could potentially be avoided with the use of a mean model on higher levels of the hierarchy.

[3] `pooh11.jpg` image obtained from publicly available database used in [109]. Website was last at `http://cobweb.ecn.purdue.edu/~jbpark/gallery.htm`.

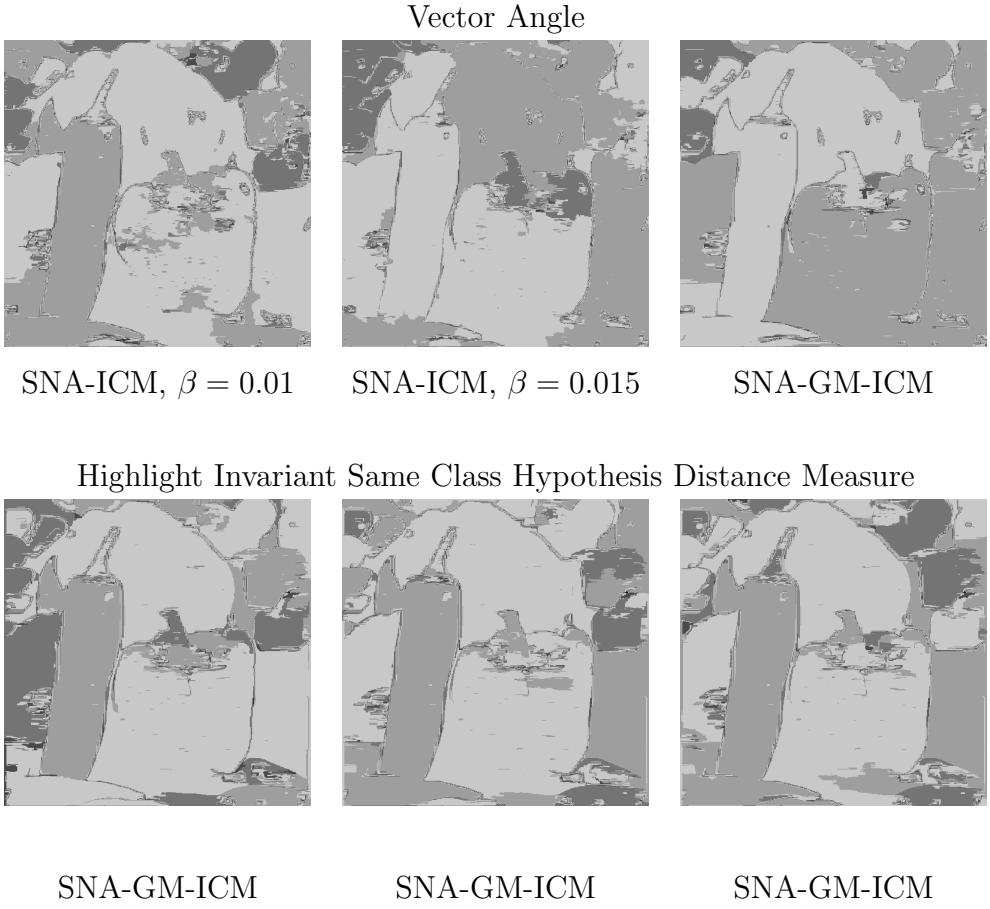


Figure 6.14: Results on the peppers image using the HI-SCH distance measure. The top row shows the vector angle distance measure obtained using SNA-ICM and SNA-GM-ICM (graduated model: $\beta = \{0.001, 0.002, 0.003, 0.004, 0.005, 0.010, 0.015\}$). Three results for SNA-GM-ICM obtained with a schedule of $\beta = \{0.025, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$. Note that virtually all highlights and dark regions have been subsumed into the surrounding regions by the new distance measure which is definitely not the case for vector angle alone (where dark regions and highlights appear as many regions). Some regions of differing color are merged together usually through the intermediary of a dark region. These images were obtained on average in 180 seconds after an average of 22 nested levels and 1.5 million site visits.

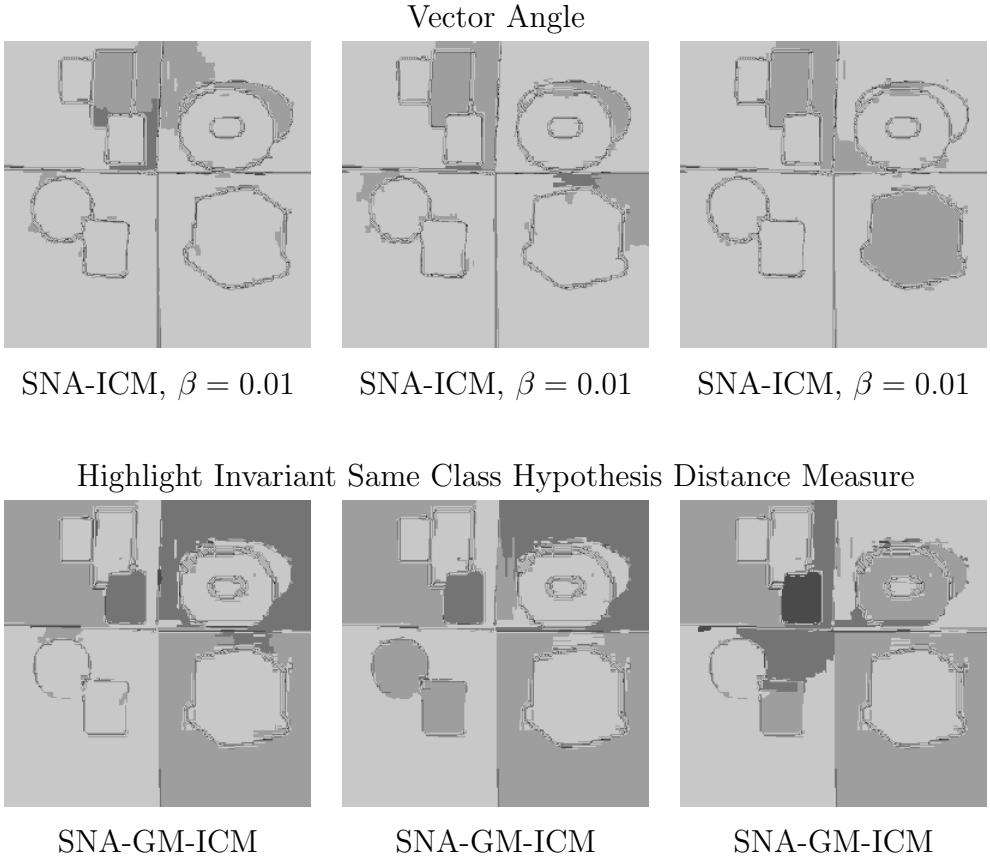


Figure 6.15: Results on the toys image using the highlight invariant same class hypothesis distance measure. Three results for SNA-GM-ICM ran with a schedule of $\beta = \{0.05, 0.1, 0.15, 0.2, 0.25, 0.5, 0.75\}$. Note that virtually all highlights and dark regions have been subsumed into the surrounding regions by the new distance measure which is also the case for vector angle. Results are also shading invariant. Note that some chromatic regions are merged because they are connected via a dark region.

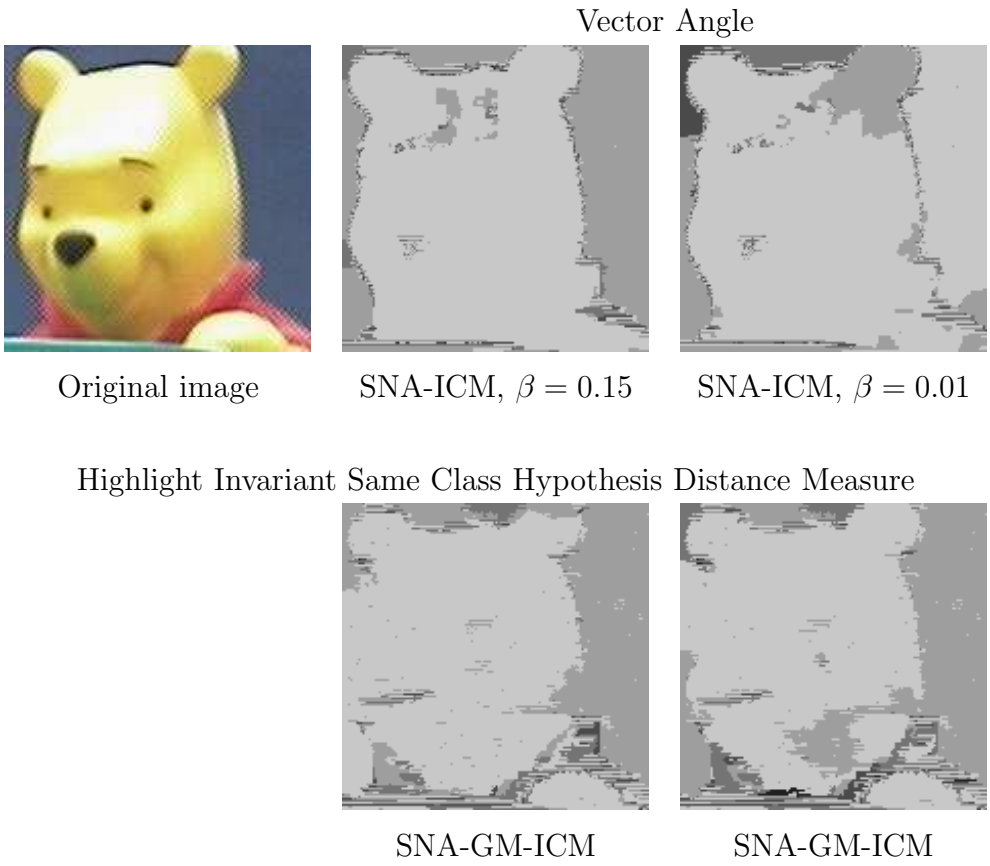


Figure 6.16: Results on the pooh image using the HI-SCH distance measure. Three results for SNA-GM-ICM ran with a schedule of $\beta = \{0.05, 0.1, 0.2, \ldots, 2.8, 2.9\}$. Note that the large highlight on the forehead of Pooh has not been subsumed into the region color regions in the vector angle result on the left. In the second vector angle result, the Pooh object merges with the background. The probabilistic distance measure does not have any problems with the highlight; however, since texture is present in the image, this leads to flawed image segmentation results.

Chapter 7

Pixel Grouping: Prototype-Based Methods

Many results in Chapter 5 are encouraging for reducing the computational burden of Simulated Annealing or Iterated Conditional Modes. We will now address a very different problem: *clustering* pixels with contextual constraints. In prototype- or clustering-based methods, computation is performed on single pixels without taking into consideration spatial information. These methods are usually set up to minimize the mean squared error of fitting the data to the prototypes [35]. This is usually done using vector quantization (VQ). VQ has many forms and depending on the end application different algorithms have been developed. Some of the better known are the k-means and its many variants [35], mixture of principal components algorithms (a vector angle-based variant of k-means) [34] (vector angle is used as similarity criterion instead of Euclidean distance), and mixture of Gaussians [35], all detailed in Section 3.2. The primary drawback of these techniques is their inability to take into account local context to avoid the formation of many small extraneous regions. Furthermore, one has to usually specify the number of classes/labels K to be used in the processing.

On the other hand, the primary drawback of region growing methods [59, 92, 137] is that they are strictly local, pixel-neighbor models and suffer from the region-to-region spilling problem: two vastly differently colored pixels may be grouped into a single region if they are linked by noisy or intermediately-colored pixels (see detailed treatment in Sections 3.3

Algorithm 9 A Vector Angle-Based Region Growing Algorithm Using Region Prototypes

1: Select seed pixels within the image.
2: **for** Each seed pixel **do**
3: Set the region prototype $\underline{w}_k$ to be the seed pixel;
4: Compute (4.12) between the region prototype and the candidate pixel;
5: Compute (4.12) between the candidate and its nearest neighbor in the region;
6: Include the candidate pixel if both similarity measures are higher than experimentally-set thresholds;
7: Update the region prototype by calculating the new principal component;
8: **end for**

and 5.3.7).

In this chapter, we develop prototype-based algorithms based on region growing and Markov Random Field-based methodologies. First, we will describe a region growing model for color image segmentation based on earlier work on grayscale images [147]. Next, a prototype-based MRF model will be described with some similarity to [107]. The chapter will conclude with results based on both methods.

7.1 Prototype-Based Region Growing

A new region growing algorithm[1] is proposed in this section based on the vector angle color similarity measure and the use of the principal component of the covariance matrix as the "characteristic" color of the region with the goal of a region-based segmentation which is perceptually-based. Our method is described in Algorithm 9.

The proposed region growing method is based on two criteria:

1. A distance threshold which ensures that adjacent pixels are similar. The distance measure used to test against this threshold could be encoded by the Euclidean distance (4.10) or any other distance measure. We propose to use the vector angle (4.12).

[1] A preliminary treatment of this approach was given in [151, 154].

2. A measure for the spread of the pixel values within a region. For grayscale images in the original paper [147], this criterion measures the spread between the minimum and maximum pixel values in a region; namely, $\max_{i \in V_r} x_i - \min_{i \in V_r} x_i$, where x_i represents the gray value of pixel i in region V_r. When the transition is made to a multidimensional pixel computing this difference becomes very expensive since we need to find the maximal difference between two vectors in an non-ordered set. The complexity of this operation is $O(|V_r|^2)$ which becomes computationally very expensive for regions with many pixels. Therefore, to limit the computational complexity the second criterion is based on the distance from the region prototype to the candidate pixel; *i.e.*, $\max_{i \in V_r}(\underline{w}_r - \underline{x}_i)$. This distance gives an indication of how close the candidate pixel is to the prototype representing the region. This criterion minimizes the mean squared error of fitting each prototype to the region's pixels.

The two region definition criteria are similar to those proposed in [146, 147]. Wang and Bhattacharya [147] gave a new definition of a connected component of a gray image which depends on two parameters: one based on the differences of the gray values of the neighboring pixels and the other based on the maximum difference between the gray values of the pixels in a region. Wang *et al.* [146] presents a region growing method based on [147] in which each region is defined by two values: the color gradient (calculated using the Euclidean distance) between two adjacent pixels and the maximum distance between two colors within this region.

Algorithm 9 presents several advantages over other region growing color image segmentation algorithms:

1. It is based on the concept of vector angle. As was shown in the case of MPC [148, 149], the vector angle is a shading-invariant color similarity measure, implying that intensity variations will be discounted in the region growing process, which is clearly not the case when using the Euclidean distance or the color spaces in [146] (the chromaticity planes u and v provide illumination independent information; however, they are used in conjunction with the XYZ color space which is illumination sensitive).

2. Since spatial information is taken into account, regions having a slightly different color, but still spatially distinct, should appear as separate regions due to the region

growing process. Furthermore, instead of computing the region prototype using the vector mean, we use the first principal component of the covariance of the data in the region thus computing the "mean" direction of the pixels in the region.

3. A final difference concerns the seed point generation. Clearly a significant disadvantage of this approach to color image segmentation is the need for seed pixels, and careful consideration needs to be given to the selection of those pixels. In [146], a complex neural network-based approach is used to determine seed pixels. Alternative approaches include finding those pixels in the color image with the greatest intensity globally, finding points with maximum local intensity or to use the MPC algorithm to select the seeds based on the clustering result. In our algorithm, the seed points are found by determining the local maximum intensity using the standard second derivative test from calculus.

Although the sensitivity to the sequence of included pixels is mitigated in Algorithm 9 through the use of region prototypes, this algorithm is still sensitive to initial seeds similarly to other region growing methods. This is a major impediment to obtaining reliable results as with different initial conditions, different results will be obtained. One way to lessen considerably the reliance of an algorithm on initial conditions is to provide a probabilistic framework such as Markov Random Fields. This is the topic of the next section. Segmentation results using Algorithm 9 are presented at the end of the chapter.

7.2 Adaptive MRF-Based Clustering

We propose to combine clustering with contextual constraints under a Gibbs/Markov Random Field modelling framework in order to apply clustering within the spatial context [107] (see Section 3.4.2 for details of past work). Therefore, we intend to find the segmented image directly as the result of energy minimization of some appropriately-defined Gibbs random field.

In [150], the authors describe a principal component analysis and vector angle clustering-based approach for color image segmentation. In this method, the prototype vector is described as the principal vector (as opposed to principal curve) of the *RGB*

color cluster and the calculation of the distance from this "cluster center" to a pixel in the image is done using the vector angle. The number of clusters is selected and the algorithm chooses the optimal (in the mean squared error-sense) multi-vector fit to the data [34]. The illumination invariances are well captured by this method, however there are several drawbacks:

1. For small (black) *RGB* values the algorithm breaks down and produces extremely noisy angles.
2. All colors must fit into a predetermined number of clusters.
3. Border areas composed of composite colors are classified arbitrarily.

The most notable drawback of color clustering methods [89, 111, 134, 148] is that they normally do not take any spatial relationships into account, and determine the segmentation strictly on a pixel-by-pixel basis, normally using the Euclidean distance. We will demonstrate for the problems of our interest, specifically the segmentation of images involving illumination effects, some degree of spatial dependence is *crucial* in formulating an adequate approach. The ability for Markov/Gibbs methods to model spatial dependencies will make them a very natural fit to our context[2].

7.2.1 Model Definition

We will formulate a color image processing and segmentation technique in the context of the Dichromatic Reflection Model [123, 145], which was introduced in Section 4.2. In this context, highlight and shading invariant color image segmentation means the finding of regions, homogenous in color, irrespective of illumination effects.

We will use the Potts model (5.10) which we reproduce here for ease of reference:

$$U(\ell) = \sum_{i,i'} \left[\alpha\Phi(x_i, x_{i'})\delta_{l_i,l_{i'}} + \beta(1 - \delta_{l_i,l_{i'}})\right] \tag{7.1}$$

where α and β control the relative constraints on the homogeneity of a single region, and the degree of region fragmentation, respectively. Model (7.1) is intuitive and easily implemented.

[2] A preliminary treatment of this approach was given in [40, 152].

The primary drawback of (7.1) is that it is strictly a local, pixel-neighbor model and suffers from the same problems as other region-growing approaches: two very differently colored pixels may be grouped into a single region if they are linked by noisy or intermediately-colored pixels. A second undesired effect is that K constrains only the number of region labels, not the number of regions; that is, in regions of noise or color-gradients, (7.1) can generate a proliferation of small regions. Finally, the label criterion, controlled by β, measures boundary length, rather than region volume (see detailed discussion of the limitations of Gibbs sampling in Section 5.1 on page 75).

An alternative approach using a global MRF model would overcome the region growing drawbacks. Pappas [107] introduced the adaptive k-means algorithm where an MRF-based refining strategy on the clustering result is done with model (3.5). Using this model allows the MRF energy to include a term penalizing the distance between a pixel and its associated prototype. In this thesis, this idea is further extended to allow the prototypes to be based on quantities related to the distance measures used between the pixel and the prototype, as well as to use continuous Gibbs sampling to obtain the prototypes themselves.

If we associate with a pixel i label l_i (or $l_{i,j}$ for pixel (i,j)) and a prototype vector $\{\underline{w}_k\}$ for a region labelled k from a set $\mathcal{L} = \{1, \ldots, K\}$ (*i.e.*, l_i corresponds to k for each pixel i) then each region is forced to be well defined:

$$U[\{l_i, \underline{w}_{l_i})\}] = \sum_{i,i'} \left[\Phi(\underline{x}_{i'}, \underline{w}_{l_i}) + \beta(1 - \delta_{l_i,l_{i'}})\right]. \tag{7.2}$$

Note that $\underline{w}_k \equiv \underline{w}_{l_i}$. Model (7.2) is a tradeoff between a completely local region growing approach, where many spurious regions can be created, and a global clustering approach where regions with differing features (such as color) can be inadvertently merged. It uses vector angle (4.12) as the distance measure Φ.

7.2.2 Color Segmentation

Model (7.1) misses one essential point: not all of the vector angles are computed with the same accuracy. Even a small amount of pixel noise on a dark or highlight region results in nearly random vector angles, which (7.1) would choose to separate into single-pixel regions. Therefore, in regions where the vector-angle criterion is vague (that is, in saturated or dark

regions), a large number of pixels may have to be flipped to see *any* change in the energy, implying that only the slowest of annealing schedules will successfully converge.

Given the covariance of the vector angle difference, computed by analytic or Monte-Carlo means [40], we introduce weights

$$u_{i,i'} = \frac{1}{\mathrm{var}(\Phi_V(\underline{\tilde{x}}_i, \underline{\tilde{x}}_{i'}))} \tag{7.3}$$

where $\underline{\tilde{x}}_i$ represents the pixels in the highlight invariant space, to assert the degree of confidence of the terms in the energy (7.1):

$$U(\ell) = \sum_{i,i'} \left[u_{i,i'} \Phi(x_i, x_{i'}) + \beta(1 - \delta_{l_i, l_{i'}}) \right] \tag{7.4}$$

Model (7.4) is a very credible segmentation criterion, representing a considerable advance beyond standard vector-angle methods, and yet (7.4) has the same drawbacks as (7.1).

The degree to which the region color is to be asserted at each pixel should be spatially-varying, now for two reasons:

1. The color-dependent effect of noise, particularly for dark and highlight pixels.

2. We are normally not interested in pixels in regions of high color gradient; at the very least, these pixels should not unduly influence the Gibbs energy by being inconsistent with the region color $\underline{w}_k$.

If we let

$$u_{i,i'} = \min \left\{ \frac{1}{\mathrm{var}(\Phi_V(\underline{\tilde{x}}_i, \underline{\tilde{x}}_{i'}))} \frac{1}{\mathrm{var}_{\mathcal{N}}(\Phi_V(\underline{\tilde{x}}_i, \underline{\tilde{x}}_{i'}))} \right\}, \tag{7.5}$$

we obtain a highlight invariant pixel reliability measure. Note that the variances are the pointwise one, based on a noise model, and a spatial one, computed over a local neighborhood $\mathcal{N}$, then our segmentation model (7.2) becomes

$$U[\{l_i, \underline{w}_{l_i}\}] = \sum_{i,i'} \left[u_{i,i'} \Phi(\underline{\tilde{w}}_{l_i}, \underline{\tilde{x}}_i) + \beta(1 - \delta_{l_i, l_{i'}}) \right] \tag{7.6}$$

This gives us a concise and coherent representation of the color image segmentation problem by incorporating both local and global constraints. The global constraints are defined by global color region labels obtained through some vector quantization process as in [150]. Local constraints are included by virtue of using pixel level constraints in the MRF model.

7.2.3 Calculating the Region Prototypes

Two different methods can be used to determine the region prototypes $\{\underline{w}_k\}$. As in previous algorithms [107], the region's prototype can be computed as the mean of the pixels or nodes in the region. For the algorithm presented here, this would mean computing the first principal component of the covariance matrix of the region pixels as this is done in MPC [34, 150]. The initial region prototypes would be initialized using the vector quantization output such as that obtained from the MPC.

The other option would be to obtain the region prototypes by sampling the distribution of all pixels within the region. Ideally, $\{\underline{w}_k\}$ would be found using continuous Gibbs Sampling. The sampling and annealing for clustering-based methods takes place not only over label indices $\{l(i,j)\}$, but also over the continuous valued region prototypes $\{\underline{w}_{l_i}\}$. However, due to the computational intractability of finding the optimal $\{\underline{w}_{l_i}\}$ in $\mathbb{R}^d$, we need to transform this continuous optimization problem into a discrete optimization problem by quantizing the possible values of $\{\underline{w}_{l_i}\}$. Note that the initial region prototypes would be assigned randomly (for faster processing they could be assigned through a vector quantization step [107]).

Recall that the Gibbs distribution for continuous values is written as:

$$P(\ell) = e^{-U(\ell)/T}/Z \tag{7.7}$$

where $U(\ell)$ is the energy being minimized, and the partition function Z is defined as

$$Z = \int e^{-U(\ell)/T} d\ell. \tag{7.8}$$

As with other Gibbs sampling approaches we do not need to calculate Z, as we limit ourselves to computing the marginal conditional probability at each pixel.

Since the conditional marginal distributions are continuous, we obtain $\{\underline{w}_k\}$ by quantizing the solution space and obtaining the marginal distribution by using (2.11) (see Section 2.4 on page 21). Thus given marginal distributions and quantization of $\{\underline{w}_k\}$, the probabilities (7.7) can be calculated. Drawing samples from the conditional marginal distribution of the Gibbs distribution implements the continuous Gibbs sampler. Algorithm 10 describes the prototype-based MRF method.

Algorithm 10 Prototype-Based MRF-Based Image Segmentation Algorithm

1: All pixel labels are randomly initialized
2: Region prototypes $\{\underline{w}_{l_i}\}$ are initialized to some vector quantized values (either k-means or Mixture of Principal Components depending on the distance measure used)
3: **for** Iterations i **do**
4: **for** Each pixel in the image **do**
5: Minimize the energy in model (7.2) or (7.6) by sampling labels using the discrete Gibbs sampler [48]
6: Sample from the conditional Gibbs distribution of the region pixels
7: **end for**
8: Lower the temperature T
9: **end for**

Applying simulated annealing to the usual 256 quantization levels present in grayscale images and even more so to 256^3 levels in RGB images is computationally prohibitive. Therefore, the region color associated with a particular label is obtained by drawing a sample from the distribution representing all the pixels that have this label. In practice, a color space could be further quantized and only coarse region features would then be used as prototypes. However, this would also take a considerable computational effort. Our practical implementation would only require that potential color regions be sampled from pixel vectors associated with that region label. Although this does not allow the use of intermediate or interpolated pixel values for the regions, in practice this is not necessary as we are trying to identify regions homogenous with respect to a particular feature with as many clusters as is necessary.

7.3 Results

7.3.1 Region Growing

Results were obtained on the fruits image shown in Figure 7.1(a). The black area represents the lack of regions since there were no seed pixels there and no regions were able to grow into those areas. Eight regions were found in Figure 7.1(b). The results clearly show that

most of the highlights have been subsumed into their respective surfaces. However, some highlights still do remain. There are two possible causes for this:

1. Parameters of the algorithm could be further adjusted.

2. The highlight areas are saturated with the illumination color.

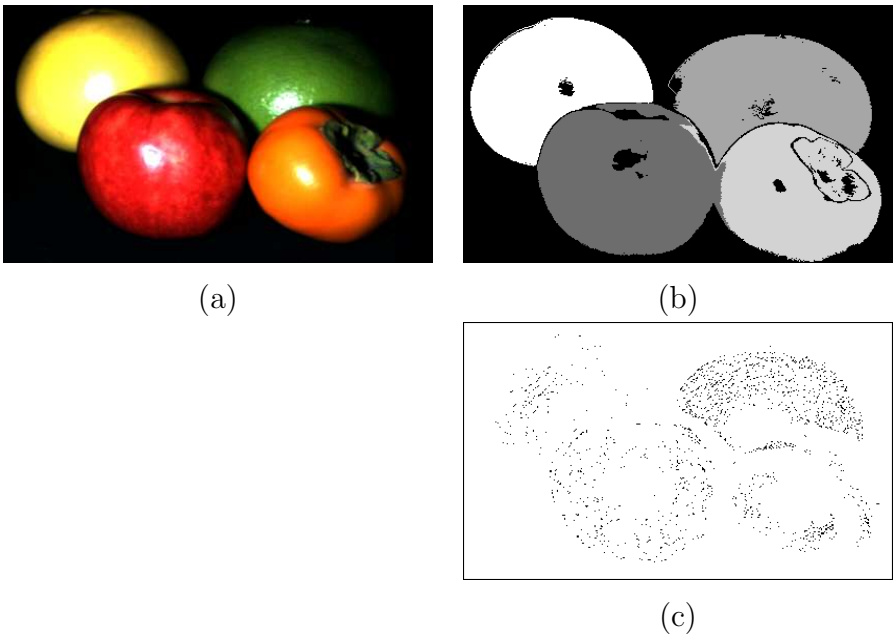
The algorithm was run with an angle tolerance of 1 on both distance measures (*i.e.*, prototype-to-pixel and pixel-to-pixel). Experimentation showed that a higher tolerance would subsume more of the highlight areas but also would cause regions to merge which are different in color (*e.g.*, the two fruit regions near the bottom of the image merged to a greater extent). The number of classes was fixed for MPC [150] and therefore all non-black pixels had to be classified as one of the regions whereas in the region growing approach only pixels satisfying aggregation criteria were included in the final partition.

A small number of pixels in the fruits image is fully saturated which results in several pixels corresponding to the zero vector due to the highlight invariance transformation (see Section 6.6). When too much light reaches the camera sensor, saturation of the image pixels with illumination light occurs. Vector angle (4.12) (see page 68) is undefined for the zero vector which explains why there are small "holes" in the regions corresponding to the location of the saturated highlights. This shows a fundamental inadequacy of the region growing paradigm. A stochastic approach such as Markov Random Fields is a much more appropriate framework for dealing with the results of between pixel vector angle computations (see Chapter 5 for details).

7.3.2 MRF Modelling

To make a comparison as straightforward as possible, all MRF results were initialized from a random start, although in practice initializing from an MPC or another segmentation result could accelerate convergence. For models (7.6) and (7.2) the label prototypes $\underline{w}_k$ are determined using the algorithm presented in [150].

Results were obtained on an artificial image of colored bands, shown in Figure 7.2(a). The artificial image varies in intensity horizontally (*i.e.*, from left to right and a saturated highlight is present near the right border). Some additive uniform uniformly distributed noise was added to this image.



(a) (b) (c)

Figure 7.1: Results for prototype-based region growing algorithm: (a) original image, (b) region growing result and (c) seeds determined using the local maximum intensity [151].

The MPC result on the artificial image is shown in Figure 7.2(b). The highlight part is clearly a mixture of the three other segmentation classes due to having a nearly zero vector representation in the $R'G'B'$ space, and the absence of spatial constraints prevents the ambiguity from being corrected. For the MRF models, the results in Figure 7.3(a) and Figure 7.3(b) clearly illustrate the problems of boundary length discussed in Section 5.1, because of the lack of region-defining constraints such as a prototype region vector, boundary length or area size constraints. It is interesting to note that under careful examination, regions generated on both sides of the border between each color band pair are not part of the same class. Figure 7.3(c) demonstrates the type of result that is obtained using (7.2). As desired, no highlight parts remain as these areas have been subsumed into their adjacent regions.

The free parameter β clearly controls the significance of the color-angle dot product in relation to the spatial label contribution in the energy term; clearly in the limit of a small value of β, the MRF result converges to that of MPC. Figure 7.3(d) shows the results for the same color bands, but now the vector angle calculation is weighted in terms of the accuracy to which the angle can be determined (which is affected by darkness or degree of highlight), as in (7.6). The main difference between models (7.2) and (7.6) seems to be the faster speed of convergence of the latter over the former.

Similar results are obtained for the adaptive prototype case and are shown in panels (e) and (f) in Figure 7.3. The main difference between these results and the previous ones was that the adaptive models were initialized using a random set of prototypes whereas the fixed adaptive models were initialized using the results of the Mixture of Principal Components algorithm. The adaptive models ran for approximately the same number of iterations on average as the fixed models. However, each iteration (*i.e.*, cycle through all points in the image) was considerably more computationally intensive in the adaptive model case.

7.4 Conclusions

A new framework for adaptive color image segmentation using Markov Random Fields and continuous Gibbs sampling has been presented. The new method presents several

(a) (b)

Figure 7.2: Color band image: (a) Original, (b) MPC segmentation.

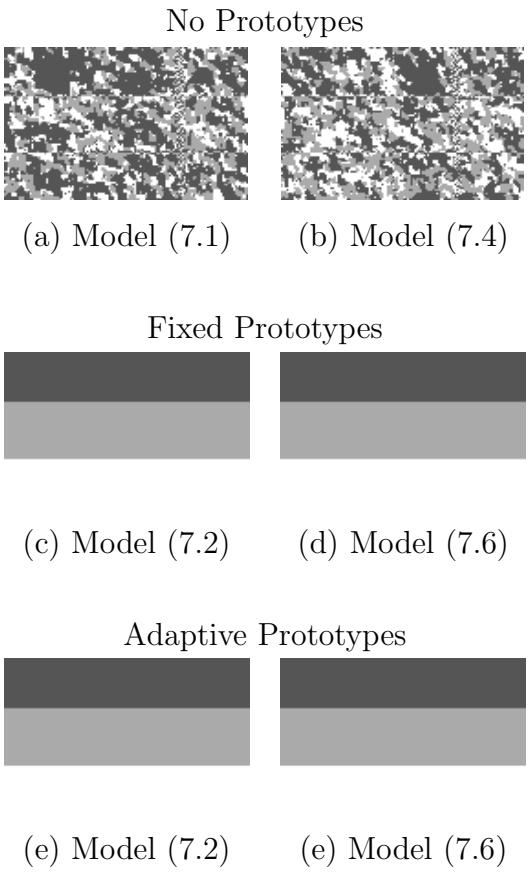


Figure 7.3: Results of prototype-based MRF models on color band image.

advantages: adaptability of global constraints (region colors) to the data, sampling over both region labels and region colors using the Gibbs sampler (both discrete and continuous), optimization of local contextual constraints (taking into account local features) with a global energy function (making sure that regions are optimally segmented with respect to each other). MRFs provide a flexible framework optimizes adaptively and globally local constraints.

Chapter 8

Pixel Comparison: Phase Unwrapping

The chapter details an original image segmentation-based contribution to the generation of highly accurate digital elevation maps through processing of interferometric Synthetic Aperture Radar (SAR) images. In the past four decades the processing of SAR images has been used extensively for terrain mapping and other remote sensing applications [16, 117]. Operating at microwave frequencies, SAR systems produce images based on the electromagnetic and geometrical properties of a surface in almost all weather conditions. By providing its own illumination, a synthetic aperture radar can be used regardless of the time of day. Thus, SAR data by themselves or together with data from other remote sensing instruments are increasingly applied to geophysical problems [117] such as polar ice research, biomass measurements, land use mapping, vegetation mapping, ocean wind estimation and soil moisture mapping.

A conventional SAR only measures the location of a target in a two-dimensional coordinate system, with one axis along the flight track or *along-track direction* and the other axis defined as the range from the SAR to the target otherwise known as the *cross-track direction*. In a SAR image, the target locations are distorted relative to a planimetric view which may lead to the incorrect interpretation of the imagery for many applications. Acquiring a SAR image is inherently a coherent process where a phase and amplitude of the radar signal correspond to an image pixel. The complex vector sum of radar echoes

from each scattering element (corresponding to a resolution cell on the ground) determines the phase of an image pixel based on the two-way range to the satellite which will vary by several hundred wavelengths across a resolution cell [82]. Thus, by itself, the phase of an image pixel appears random.

Interferometric SAR or inSAR data is fundamentally different from SAR data as it is based on two SAR images such that given a repeat acquisition, a correlation between the phases of corresponding image pixels exists. There are two possibilities: either the repeat acquisition is made at the same time with two sensors in space (resulting in images with the same orbital geometry and constant ground scattering characteristics) which is the case for ERS-1/2 [57] or the acquisition is made using the same radar at two different times (resulting in images having parallel spatially separated orbits and most likely different ground scattering characteristics), for example RADARSAT [56, 108]. For the first case, phase values are correlated and the phase shift corresponds to the difference in range. In the second case, the time difference in acquiring the images causes a loss of coherence which makes the images more difficult to process [100].

The phase shift at the scale of the overall image corresponds to an interference pattern which is a function of both orbital geometry and surface topography. Knowing the orbital geometry, it is then possible to infer the surface topography [117]. Therefore, the basic idea behind SAR interferometry is that three dimensional data can be extracted from the interferometric pattern based on the complex SAR image pair. This leads to much more accurate digital elevation models with such applications as cartography and change detection which is very useful for studying earthquakes and other tectonic movements of the Earth which is possible when both the orbital geometry and surface topography are known. The quality of correlation between two SAR acquisitions can also be determined and is known as the coherence. Coherence indicates how much correlation there is in the phase data. For example, coherence is low for areas where the phase changes abruptly such as significant changes in elevation. The phase plane gives us the modulo 2π phase information (*i.e.*, the range is $[0, 2\pi]$) while the coherence is a measure of the reliability of the phase value (given in the $[0, 1]$ range).

InSAR involves the derivation of topographic information from the radar phase which can then be used for digital elevation model (DEM) generation [16]. Elevation is pro-

portional to the full phase, whereas the measured phase is modulo 2π, necessitating a well-defined process to recover the full phase values via fringe-counting or phase unwrapping. The relationship between the measured and full phases is given by

$$\phi_i = P_i + 2k_i\pi \tag{8.1}$$

where ϕ_i is the full phase and P_i is the measured phase at pixel i. The main problem resides in finding the integer k_i in order to reconstruct the original phase ϕ_i. If phase measurements are noise-free, this can be easily done as long as there is no steep topography. However, decorrelation noise, atmospheric distortions and size of the collected images make actual measurements more challenging to process [117]. The main unwrapping problems occur in noisy areas or in areas of high elevation where the phase is compressed into narrow bands and cannot be easily unwrapped. In general, phase unwrapping is a very difficult inverse problem as we cannot make any prior assumptions about image location (other than that a transition occurs between the modulo 2π measured phases) since we could be measuring any part of the Earth. Even if we knew exactly the position, the topography of the land might have changed due to tectonic movements and therefore we cannot condition our results based on prior measurements.

Figure 8.1 shows a typical measured phase and coherence signal. Observe that the phase image has bands of pixels with well delineated borders in most areas. Since the phase is measured modulo 2π, the point at which there is a sharp edge between two phase values corresponds to the transition between two fringes. Given the prominence of edges between measured phases, the separation of these phase bands could be done using an image segmentation algorithm [162].

In this chapter, we will adopt an image processing framework based on Stochastic Nested Aggregation (SNA) and the Potts model developed in Chapter 5 in order to solve the phase unwrapping problem. In phase unwrapping, phase discontinuities are very prominent with respect to all other pixel-to-pixel transitions and can be easily modelled as edges. The Potts model is a natural choice to model those discontinuities since its most basic form is based on pixel differences and not region characteristics. Once the phase image is segmented, the relative topographic ordering between regions can be computed in order to find the k integers that are required to reconstruct the original phase. In addition to

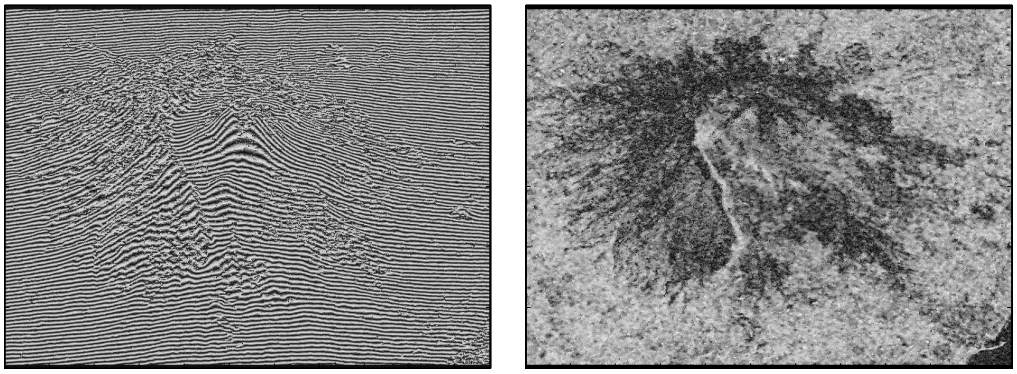


Figure 8.1: An inSAR image pair showing Mt. Vesuvius: original phase image (left), and original coherence image (right). the phase image shows bands with phase values between 0 (dark pixels) and 2π (light pixels). The coherence image shows which phase values can be trusted (light pixels) and which are uncertain (dark pixels).

using the cost function of [17], we also develop an approximate cost function to drive the segmentation and evaluate its performance.

This chapter is organized in the following manner. Section 8.1 presents an overview of current and past methods for phase unwrapping. Section 8.2 reviews hierarchical models which were used for the processing of inSAR images. Section 8.3 summarizes a probabilistic cost function with a non-linear relationship between phase and coherence which was used in a network flow paradigm [17]. Section 8.4 describes the new phase unwrapping model. Section 8.5 summarizes the principal results obtained using the cost function of [16] and the new cost function. The final section concludes the chapter.

8.1 Literature Review

There are two main approaches to phase unwrapping. The first class of algorithms is based on the integration with branch cuts approach initially developed by Goldstein *et al.* [53]. A second class of algorithms is based on a least-squares (LS) fitting of the unwrapped

solution to the gradients of the wrapped phase [51]. We will also review approaches based on network flow and image segmentation separate from branch cuts given their current significance. For a comprehensive summary of various approaches see [52, 117].

8.1.1 Path Integral Methods: Branch Cuts

One approach to phase unwrapping would be to calculate the first differences of the phase at each image point in the vertical and horizontal directions as an approximation to the derivative. The result can then be integrated. By directly applying this approach local errors due to phase noise are propagated across the full SAR image [53]. To mitigate the propagation of error, many branch-cut algorithms unwrap the phase via paths that lead to self-consistent solutions thus trying to isolate error sources prior to integration [53]. The first step is to calculate the phase differences in order to map them into the interval $[-\pi, \pi)$. The assumption is that the true unwrapped phase does not change by more than 2π between adjacent pixels. Unwrapping errors occur due to inconsistencies introduced when this assumption is violated (*e.g.*, because of statistical variations or abrupt shifts in the true phase).

In the error-free case, the integral of the differenced phase should be zero since the solution should be independent of the path of integration (note that the integration is done about a closed path formed by four mutually neighboring set of pixels). Therefore, nonzero results indicate phase inconsistencies. These points are called *residues* and are assigned positive or negative *charges* depending on the sign of the sum (by convention summing is done in a clockwise manner). Integration results in the sum of the enclosed residues. Consequently, non-zero residues must be avoided by connecting, for example, residues with opposite charges through branch cuts that the path of integration cannot intersect [52].

An interferogram may have a slight net charge which can be mitigated with a connection to the border of the interferogram. Once branch cuts have been selected, integrating the differenced phase (given that paths of integration do not cross branch cuts) finishes the phase unwrapping.

The key characteristic of branch cut algorithms is how the selection of branch cuts is achieved [117]. In most cases, the number of residues is very high making the evaluation

of the totality of possible solutions a computationally impossible task. Thus, heuristic methods have usually been adopted for branch cut selection algorithms in order to limit the search space [52, 53].

8.1.2 Path Integral Methods: Network Flow

Network flow is used to convert the phase unwrapping problem into a discrete optimization problem where the global cost of loop integrals is being minimized [16, 17, 23, 29, 30, 36]. The network flow problem is defined as a set of nodes, a set of arcs connecting the nodes, a supply-demand function on each node and cost functions associated with each node [17]. Two conditions must be satisfied: the sum of all flow supplies and demands must be zero and at each node the total outgoing flow must be the same as the total incoming flow plus the flow generation within the node. The flow on each arc represents the residual (*i.e.*, the number of 2π multiples between the wrapped and unwrapped phase gradients) and is the quantity that we would like to discover.

Residuals are defined as [30]:

$$k_q = \frac{1}{2\pi}\left(\phi_i - \phi_{i'} - (P_i - P_{i'})\right) \tag{8.2}$$

where q is an arc between nodes i and i'. Then the phase unwrapping problem is given by

$$\min \sum_q c_q |k_q| \tag{8.3}$$

where c_q represents the nonnegative confidence weights on the residuals. The residue loop integrals are set up using a 2×2 window. Therefore given pixels a, b, c and d (with a, b in the top row and c, d in the bottom row of the 2×2 matrix), we have the following loop integral constraint: $k_{ab}+k_{bc}-k_{cd}-k_{da} = \frac{1}{2\pi}\left(P_{ab} + P_{bc} - P_{cd} - P_{da}\right)$. Now if we assume that $x_q^+ = \max(0, k_q)$ and $x_q^- = \max(0, -k_q)$ where $x_q^+ \geq 0$ and $x_q^- \geq 0$, then we can rewrite the problem as a minimum cost network flow by using $k_q = x_q^+ - x_q^-$ and $|k_q| = x_q^+ + x_q^-$. The residues are given by applying the difference operator to a phase image while $c_q^{\pm}$ are given by a cost function (see Section 8.3).

The network flow mechanism works very well and methods differ by how they calculate the cost function [16, 23, 29, 30, 36]. Our method is similar to network flow in that a global

energy is being minimized locally. The difference between the methods is that we need *not* unwrap the whole image in one step, rather, using nested aggregation the unwrapping can be done gradually in a hierarchy of image segmentations.

8.1.3 Path Integral Methods: Image Segmentation

In the image segmentation domain, a region growing approach was used for phase unwrapping [162]. Their algorithm unwraps the phase by using phase information from neighboring pixels to predict the correct phase of each new pixel to be unwrapped. In addition, they apply a reliability criterion to verify each unwrapping attempt. In order to unwrap the phase via the most robust path, this criterion is slowly relaxed. Finally, regions are sometimes allowed to be merged as they grow into one another.

However, region growing methods as detailed in Section 3.3 are highly dependent on initial conditions and the sequence of pixels merged into the respective regions. Therefore, it is difficult to assess their performance since results could be widely different based on the implementation used.

8.1.4 Least Squares

The least squares-based methods provide an alternate set of phase unwrapping approach. These algorithms minimize the difference between the gradients of the solution and the wrapped phase in an LS sense [51]. The LS problem may be formulated as the solution of a linear set of equations. However, for typical image dimensions, the matrix is too large to obtain a solution by direct matrix inversion. A computationally fast and efficient solution, however, can be obtained using a fast Fourier transform (FFT) [51]. The unweighed LS solution is sensitive to inconsistencies in the wrapped phase (*i.e.*, residues), leading to significant errors in the unwrapped phase. A potentially more robust approach is to use a weighted LS solution. In this case, an iterative computational scheme (based on the FFT algorithm) is necessary to solve the linear set of equations, leading to significant increases in computation time.

Strand *et al.* improve on the weighted LS solution using a block least-squares (BLS) method [126]. This approach tessellates the input image into small square blocks with only

one phase wrap. These blocks are then unwrapped and merged together. By specifying a suitable mask, objects of any shape can be processed. The BLS method is shown to be superior quantitatively and qualitatively on synthetic images with different noise levels.

8.1.5 Discussion

There are significant differences between least squares and branch cut algorithms. Branch cut algorithms usually isolate high residue density areas from low ones resulting in holes in the unwrapped solution. Errors in a branch cut solution are localized and always integer multiples of 2π with a result consisting of two types of regions: correctly unwrapped regions and regions with an error that is an integer multiple of 2π. On the other hand, LS algorithms produce continuous solutions even where the phase noise is high potentially introducing large-scale errors since errors are continuous and distributed over the entire solution.

Phase unwrapping using path integration is a well established method which has been applied to a large volume of interferometric data [17, 23]. On the other hand, unweighed LS algorithms are not sufficiently robust for most practical applications [117]. Weighted LS can yield improved results; however, these results are highly weights-dependent. Selecting weighing coefficients is a problem of similar complexity to that of choosing branch cuts [117].

Given the relative success of path integral methods, we will purse this strategy using the stochastic nested aggregation framework. A Markov Random Field approach is applicable in this case given that the Potts model is ideally suited to tackle the phase unwrapping segmentation problem. This is because the Potts energy in its most basic form (see Section 2.5 on page 21) models edge discontinuities and not region characteristics. In phase unwrapping, phase discontinuities are very prominent with respect to all other pixel-to-pixel transitions and can be easily modelled as edges.

8.2 Hierarchical Methods for Phase Unwrapping

To find a phase unwrapping solution often entails a substantial computational effort. It is not unusual to have to process very large images (*e.g.*, in excess of 10^7 pixels) which makes

the problem very challenging. To speed up phase unwrapping, hierarchical methods can be used. For example, Carballo and Fieguth [16, 17] developed a hierarchical network flow method based on a *divide-and-conquer* approach (which could also be easily adapted to other algorithms such as those for image segmentation and branch cuts). This bottom-up irregular hierarchy-based method has been described earlier in Section 3.5.3.

In summary, the first step subdivides the image into smaller phase unwrapping subproblems (*e.g.*, 100×100 pixel blocks) which are then combined in the second step. Second, the combination step itself can be interpreted as an unwrapping problem, to which they apply a modified network flow solution. This formulation allows images of virtually unlimited size to be unwrapped leading to decreases in the algorithm execution time and memory requirements. This method is considerably slower than our proposed approach due to the initial subdivision of the image into large blocks. In stochastic nested aggregation, we can choose to create initially very small regions (*e.g.*, two pixels in size). This allows us to proceed to segment regions in a geometric fashion achieving $O(N)$ computational complexity which is much faster than most other phase unwrapping methods and just as fast as graph cuts.

Another hierarchical approach involves using the multigrid algorithm [55] in a least squares formulation [43, 113]. One algorithm carries out phase unwrapping using the weighted least squares method optimized using multigrid Gauss-Seidel relaxation [113]. By transferring the problem to ever coarser grids, the multigrid algorithm relies on transforming the low frequency components of the errors into high frequency components in order to remove them. Fornaro *et al.* devised a finite element method using an efficient multigrid implementation [43]. The technique produces a least squares solution in the time domain introducing weighting functions without increasing computational requirements. The computational speed is the main advantage of the multigrid-based techniques; however, least squares methods produce undesirable large-scale errors. Our method, being based on image segmentation introduces errors based on region-to-region spilling which produces localized errors where the result consists of correctly unwrapped regions and regions with an integer multiple of 2π error. Therefore, in contrast to least squares methods where there is always some error present, our method should produce results in most areas with zero error.

8.3 Review of Phase Unwrapping Cost Functions

Several models or cost functions have been used in phase unwrapping [17, 29, 117]. The simplest model is a function of phase discontinuities [29]. The distance measure between two adjacent pixels would then be summarized by

$$\Phi_p(\phi_i, \phi_j) = \phi_i - \phi_j \tag{8.4}$$

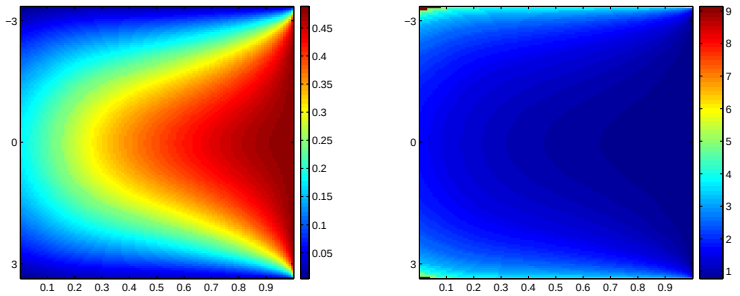
where $i \in \mathcal{N}_j$ and $j \in \mathcal{N}_i$ for all i and j. However, in inSAR phase unwrapping where the coherence map is also available, it is advisable to update the model to include some measure of confidence in (or the cost of) phase measurements.

Costantini later used the coherence values as weights for the phase differences [30]. Ghiglia and Pritt thresholded the phase slope variance map to obtain a cost function [52]. Eineder *et al.* obtained a binary cost map by thresholding the amplitude, charge density and flatness [36]. Chen and Zebker thresholded each of the coherence map and the edge detection map of the interferogram magnitude to decide which phase differences to trust [23]. Note that all of these methods are ad-hoc.

Carballo and Fieguth formulate the phase-unwrapping problem as a maximum likelihood (ML) estimation based on phase statistics [16]. They estimate the probability of phase discontinuities based on coherence and topographic slope. Specifically, they base their derivation on the probability density function for single-look and multi-look interferometric phase distributions. Their cost function is based on the probability of a zero residual whose values are shown in Figure 8.2 (they also require the probabilities of "+1" and "-1" residuals). Since this is the most rigorous specification of a relationship between phase differences and the corresponding coherence, we will later use it in our Potts model in order to segment SAR interferograms.

8.4 Phase Unwrapping Using the Potts Model

Phase unwrapping can be formulated as an image segmentation on the interferogram due to the ease of modelling edge discontinuities using the Potts model. The individual phase regions (corresponding to the k_i integers) will be individually grown from the finest level



$p^{(0)}$ $\quad\quad\quad\quad\quad\quad\quad\quad$ $-\ln p^{(0)}$

Figure 8.2: Phase unwrapping cost functions: $p^{(0)}$ is the probability of a zero residual as given by [16]. The horizontal axis represents the coherence values while the vertical axis indicates the phase differences in the $[-\pi, \pi]$ range. An absolute difference in phase of more than π is automatically considered to be a phase discontinuity.

(pixels) to the coarsest (segmented image) through a hierarchy of intermediary regions. At the finest levels, we will merge together only the phases which have little or no difference between them (*i.e.*, at least $|P_i - P_{i'}| < \pi$) modulated by the confidence we have in those phases based on the coherence signal. We will use the Potts model (5.11) (see page 100) to model the segmentation process given a discontinuity measure or cost function that we describe below.

8.4.1 New Phase Unwrapping Cost Function

Consider the probabilistic model in Figure 8.2. Observe that there is a dependence between coherence and phase difference that is non-linear; namely, that as the phase difference increases and the coherence decreases, there is a higher likelihood that there is a discontinuity. Thus the model only merges those parts of the image which are the most reliable in order to avoid problems with noise and jumping across phases. Our proposed model will mirror this development.

The distance measure devised by [16] is difficult to compute. An approximation of

the probability of zero residual might be sufficient to produce desirable phase unwrapping results. We propose the new cost function or distance measure as

$$\Phi_m(i,j) = (1 - \min(C_i, C_j))^w + \upsilon\Phi_E(\phi_i, \phi_j) \tag{8.5}$$

where C_i is the coherence at pixel i and Φ_E is the Euclidean distance between phases ϕ_i and ϕ_j. υ is a weighing parameter such that $\upsilon < \frac{1}{2}$. w is the exponent which controls the shape of the model; we set $w = 2$ in most of our experiments. Figure 8.3 shows pre-computed look-up tables for distance measure (8.5) and different values of υ and w. Note the difference in behavior: as we increase w the function tends to accept less reliable pixels for the same phase difference. As we vary υ, the proportion of the phase component becomes more prominent which means that coherence will play a smaller part in determining the phase discontinuity.

For model (8.5) with $w = 2$ and $\upsilon = 0.008$ the range of coherence values is between $[0, 1]$. The phase difference component will have a range of approximately $[0, 0.05]$. This means that largely the segmentation will be dependent on the coherence. If the two pixels are not coherent "enough," the difference between them will be deemed large. As we increase w we start trusting phase differences more since $(1 - \min(C_i, C_j))^w$ decreases.

The transition equations take the same forms as (5.12) and (5.13). When pixels are aggregated into regions, the distances Φ_m are summed together to obtain a cumulative phase aggregation criterion. Since this distance measure is based on four quantities (two coherences C_i and C_j and two phases ϕ_i and ϕ_j), it is a semi-metric as the triangle inequality does not apply (this is especially true since the metric would be different for different coherence values).

8.4.2 Unwrapping Segmented Regions

Once image segmentation has been carried out and individual phase regions have been identified, the phase of each of those regions needs to be unwrapped relative to its neighbors. The phase should be unwrapped according to how reliable or coherent the connection is between the adjacent regions. In other words, regions that are connected with highly coherent edges should be unwrapped first because they exhibit the highest level of confidence among the pixels in the image. The reliability of an edge is determined by first

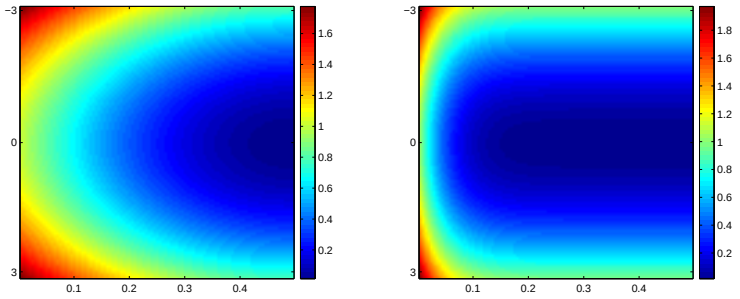


Figure 8.3: Plots of pre-computed look-up tables showing the probability distributions used to compute the costs using the new phase unwrapping cost function (8.5). The left plot shows values for phase differences (maximum magnitude of π) vs. minimum coherence for $w = 2$ and $v = 0.08$. For comparison, we also show a cost function with $w = 10$ and $v = 0.01$ on the righthand side. Note that through w and v we can effectively control the shape of the function and approach the optimal representation shown for $p^{(0)}$ in Figure 8.2.

calculating the minimum between-pixel coherence $\min(C_i, C_j)$.

The between-region coherence coefficient is then determined by summing the minimum between-pixel coherence across regions edges by using a formulation similar to the transition equations in Section 5.3:

$$B_{r,r'}^{(s+1)} = \sum_{t \in V_r^{(s)}} \sum_{t' \in V_{r'}^{(s)}} B_{t,t'}^{(s)} \tag{8.6}$$

where $B_{r,r'}^{(0)} = \min(C_i, C_j)$. This measure of region edge reliability was chosen as it permits the use of both short reliable edges or very long unreliable edges (in this case since the edge is long, it is assumed that it is collectively reliable since a long edge is highly unlikely). The edges between all regions are sorted with respect to the BRC parameter and processed in accordance to their ranking. This process is illustrated in Figure 8.4. Notice that regions with a highly reliable coherence are merged first creating "islands" of unwrapped phase. As the unwrapping proceeds, these unwrapped "islands" are merged together. This process assures the user that the phases are unwrapped only via the most reliable of paths.

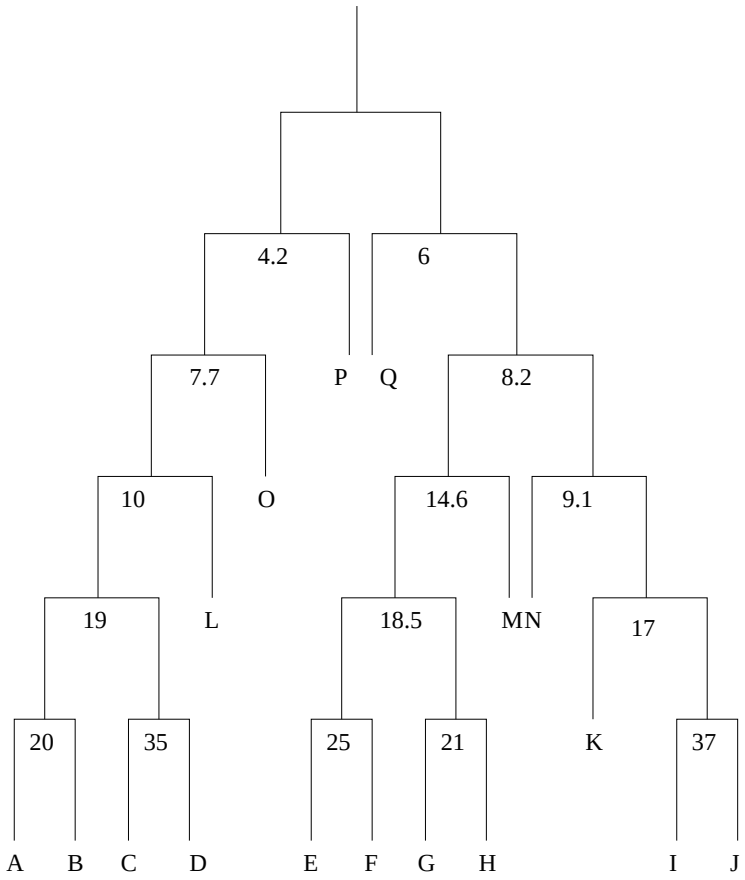


Figure 8.4: An illustration of the phase unwrapping process after the measured phase image has been segmented. Segmented phase regions (delineated by letters of the alphabet) are unwrapped with respect to their adjacent nodes in order of decreasing total between region coherence (8.6) (since we want first to unwrap regions which have a very reliable edge between them). Therefore, in the diagram the regions will be unwrapped in the following order: IJ, CD, EF, GH, AB, (AB)(CD), (EF)(GH), K(IJ), etc.

8.5 Results

We used the edge Potts model (5.11) in order to model the segmentation process. The simple edge model is the ideal model for phase unwrapping as we are only interested regions where the transitions between measured phases are high creating natural boundaries for image segmentation whereas regions where phase differences are low are of no interest. Here a mean Potts model would not make sense since regions between modulo 2π phase transitions are not homogenous (by definition) with respect to the phase.

The threshold for total edge coherence (*i.e.*, the sum of all minimum adjacent pixel coherences for an edge of arbitrary length between two regions) was set experimentally at 0.25. Performance evaluation is easily done for the synthetic images since a digital elevation model (DEM) was provided in each case. We use Algorithm 11 to evaluate the percentage of pixels h_{max}/N that are correctly labelled. Since the difference between the largest correctly unwrapped region and the truth elevation might not always be zero, an algorithm was needed to determine the largest self-consistent area that is unwrapped correctly.

Algorithm 11 Phase Unwrapping Evaluation Algorithm

1: Consider a reference DEM, X_{ref}, and the produced DEM, X_{res};
2: $X_{diff} = X_{ref} - X_{res}$;
3: Create a histogram of the values in X_{diff};
4: Identify the histogram bin with the highest number of elements, h_{max};

We will test both models shown in Figure 8.3, as well as the cost function shown in Figure 8.2. These tests will give us an idea of the power of image segmentation for unwrapping inSAR phases. Results are obtained using the SNA-ICM algorithm (see Section 5.5 on page 125), and SNA-GM-ICM was run with two iterations at each level in order to minimize computational cost. SNA-SA and SNA-GM-SA were also tested although results were not significantly different to warrant their presentation. Additionally, simulated annealing-based processing was considerably more computationally expensive than using ICM.

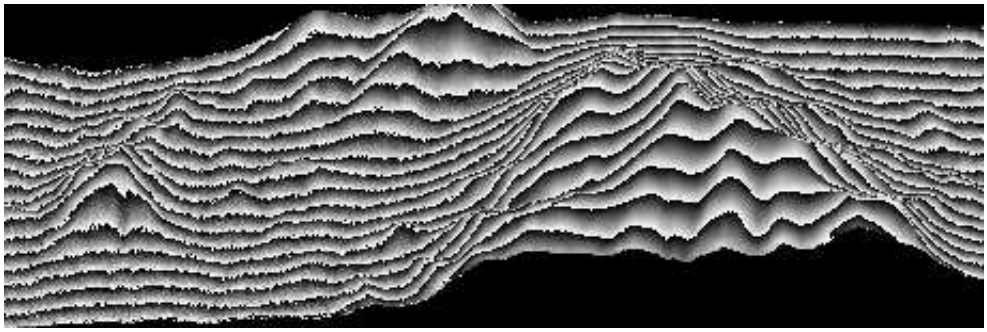
8.5.1 Simulated Data Set

Figures 8.5 show a simulated measured phase image (Long's Peak) together with a coherence image based on correlation values. The image segmenter was run with Carballo's zero residual model [17] with $\beta = \{0.5, 0.55, \ldots, 1, 1.05\}$. The resulting segmented images with the corresponding DEMs and difference images against the true elevation surface data are shown in Figures 8.6. A second test was performed with model $w = 2$ and $\upsilon = 0.4$ where a high value was needed for υ due to a large amount of noise in parts of the Long's Peak image. The corresponding β schedule was specified as $\{0.5, 0.6, \ldots, 1.2, 1.3\}$. The results for this test are shown in 8.7.

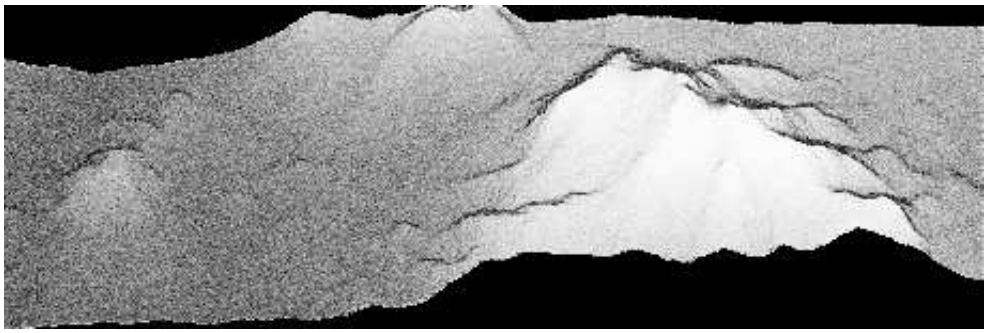
The results for Long's Peak show a credible image segmentation and DEM reconstruction using the probability of zero residual [17]. The average error rate for a set of ten experiments was 14% which is mostly due to region spilling (especially areas in the top left corner and bottom right corner where very sharp transitions occur). This is worse than Carballo *et al.*'s network flow result with an average error of 4% for the Long's Peak image [17].

However, SNA with a complexity of $O(N)$ is able to solve the problem much faster than network flow. Network flow can be solved with varying complexity depending on the algorithm used. For example the Edmonds-Karp algorithm has complexity $O(|V| \cdot |E|^2)$ (where $|V|$ or N is the number of nodes in the graph and $|E|$ defines the number of edges) and guaranteed convergence [28]. Another alternative is the Ford-Fulkerson algorithm with a complexity of $O(|E| \cdot maxflow)$ which is dependent on the maximum flow value in the graph (which could be large); in addition, it is not guaranteed to converge for non-integer flow values [28]. SNA does not have these limitations and could be applied to network flow in the future.

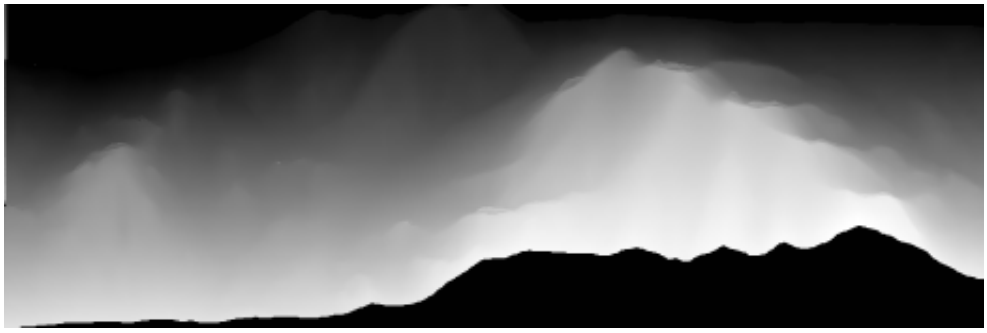
Figure 8.7 shows one result running model (8.5) with $w = 2$, and $\upsilon = 0.4$. Error rates are also on average of 12% (average of ten runs). Since some regions extend diagonally through single pixel connections (see bottom right side of image), they cannot be grouped as single regions. This is a direct consequence of using a first order neighborhood (*i.e.*, a four pixel neighborhood) in the model which does not allow diagonal relationships. Note that $\upsilon = 0.4$ and not $\upsilon = 0.08$ as was set for the Mt. Vesuvius image (see below). If we set $\upsilon = 0.08$, results would be drastically different with little phase unwrapping done in



Original phase

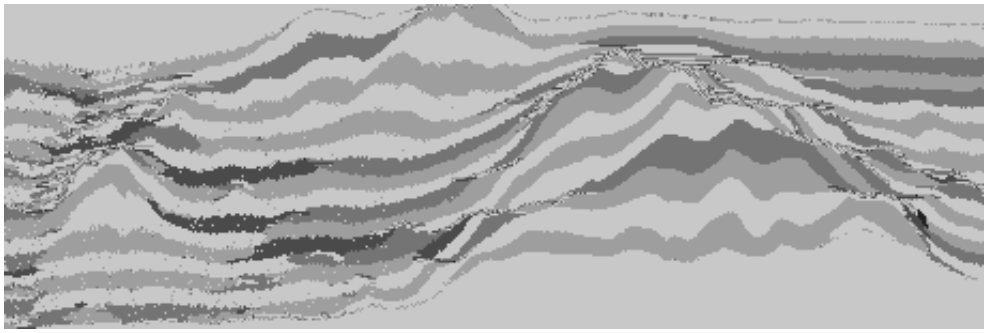


Original coherence map



Original digital elevation model

Figure 8.5: Set of simulated data for Long's Peak, Colorado. The data were obtained by simulating measured phases from true surface elevation data of Long's Peak and were obtained from [52].

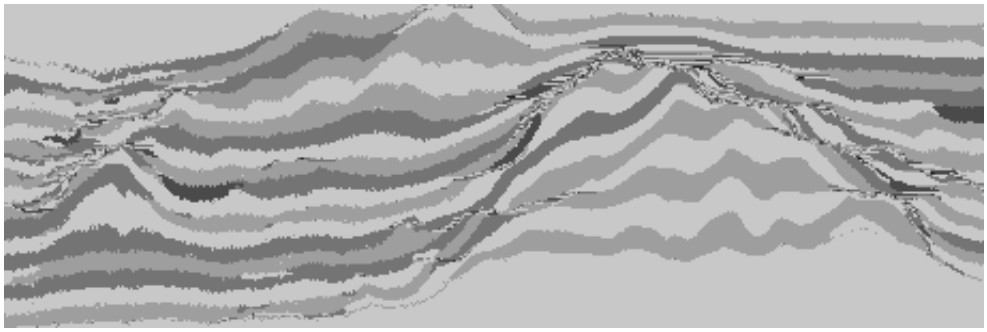


Segmentation result

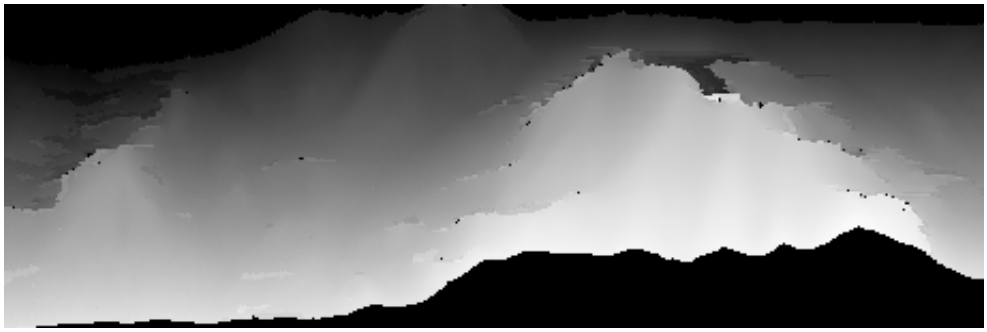
Measured digital elevation model

Difference between result DEM and ideal DEM in Figure 8.5

Figure 8.6: Results on the Long's Peak image using the probability of zero residual [16]. Note the discontinuities throughout the resulting image. These usually correspond to mountain peaks where there are compressed phase transitions due to sharp elevation changes. In addition, the area next to the left image border is very noisy leading to an incorrect segmentation.



Segmentation result



Measured digital elevation model

Difference between result DEM and ideal DEM in Figure 8.5

Figure 8.7: Results on the Long's Peak image using model (8.5) with $w = 2$ and $\upsilon = 0.08$. Note that the region in the top left corner is very noisy and leads to incorrect results.

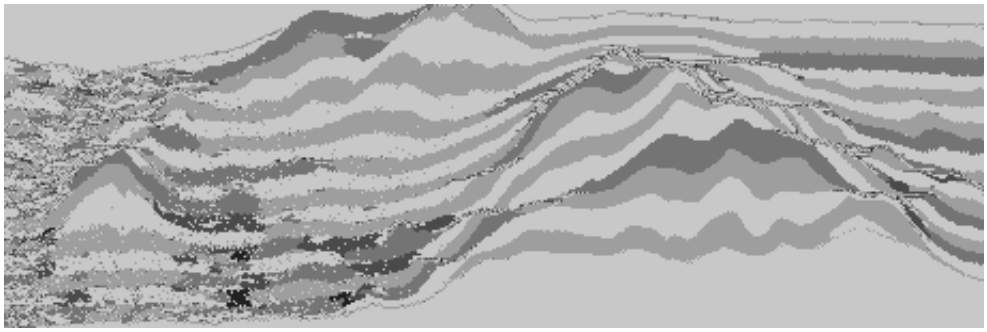
the upper left corner of the image due to high levels of noise as shown in Figure 8.8 with an increased error rate of 24%.

The main problem with image processing based techniques for phase unwrapping is region spilling. Most of the region spilling occurs through single pixel connections of large regions. While a lot of these spills can be avoided using SNA-GM-ICM and SNA-GM-SA, some spills unfortunately cannot be avoided in the current setting. Since the mean model is not available to us due to the non-homogeneity of each region, a more robust model will be needed to rival network flow in terms of error rates. Errors due to image segmentation spread very easily throughout the image with just a few mis-segmented pixels. It would also be interesting to apply network flow to phase unwrapping problems within the SNA framework.

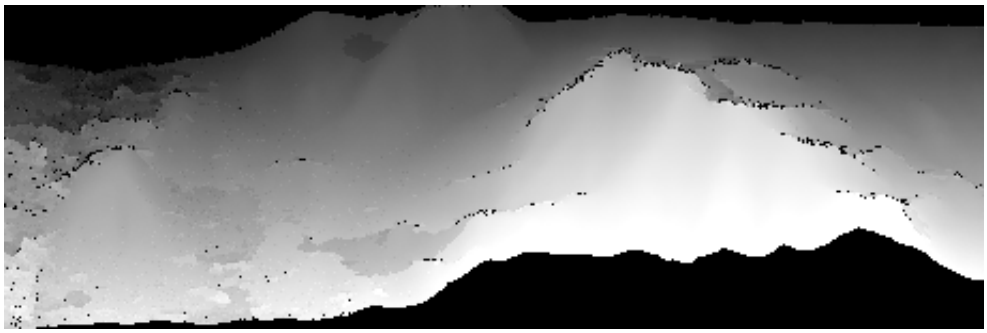
8.5.2 Real Data Set

Results are presented on the Mt. Vesuvius data shown in Figure 8.1. Here the true elevation map is not known; therefore, we cannot evaluate the results quantitatively and must limit ourselves to a qualitative analysis. The segmentation done according to the zero residual of [16] is presented in Figure 8.9. The segmentation appears plausible with very few regions that are problematic (transition from light blue to deep blue in the top left corner has a discontinuity). Other areas appear to be correctly unwrapped including the middle bottom area.

The segmentation of Long's Peak's phase image would suggest that the model to be used on other inSAR image pairs is $w = 2$ and $\upsilon = 0.4$. However, this model caused extensive region spilling in Mt. Vesuvius with a resulting undesirable digital elevation model. The model chose to segment the Mt Vesuvius interferogram was $w = 2$ and $\upsilon = 0.08$. This lead to the visually appealing segmentation and digital elevation model in Figure 8.10. Notice that the main problem areas are in the middle bottom area. Figures 8.11 and 8.12 are also shown in order to show how β influences the segmentation result. A lower β is more conservative and allows less region spilling. However, this also results in many small regions proliferating especially in areas of low coherence. A higher β allows more merging which in some areas is excessive. In this case, very few pixels remain unassigned; however, many areas although unwrapped show discontinuities. This is because of the region spilling



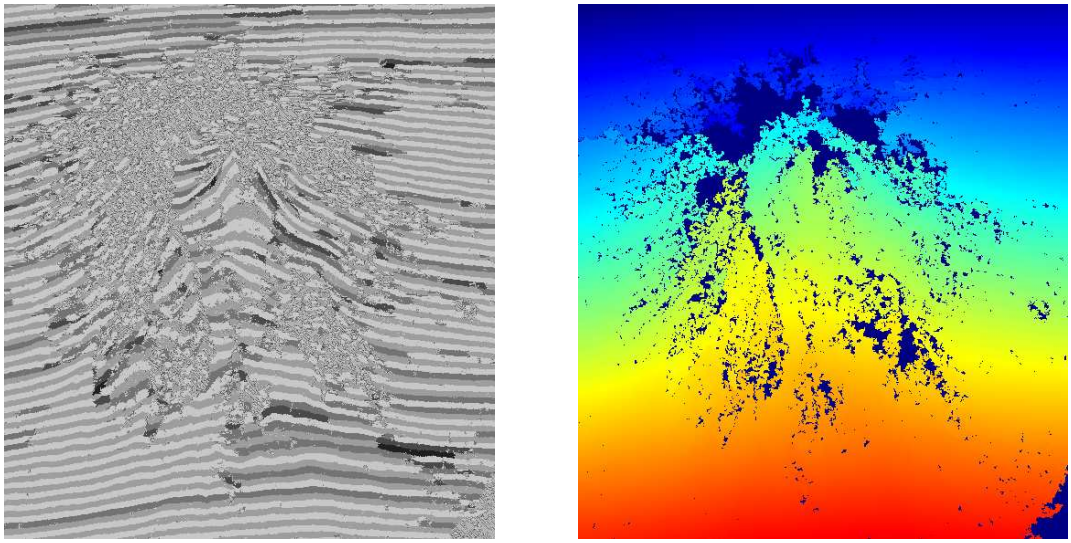
Segmentation result



Measured digital elevation model

Difference between result DEM and ideal DEM in Figure 8.5

Figure 8.8: Results on the Long's Peak image using model (8.5) with $w = 2$ and $\upsilon = 0.08$. Due to the undesirable segmentation especially in the left half of the image, results show an elevation map with many errors.



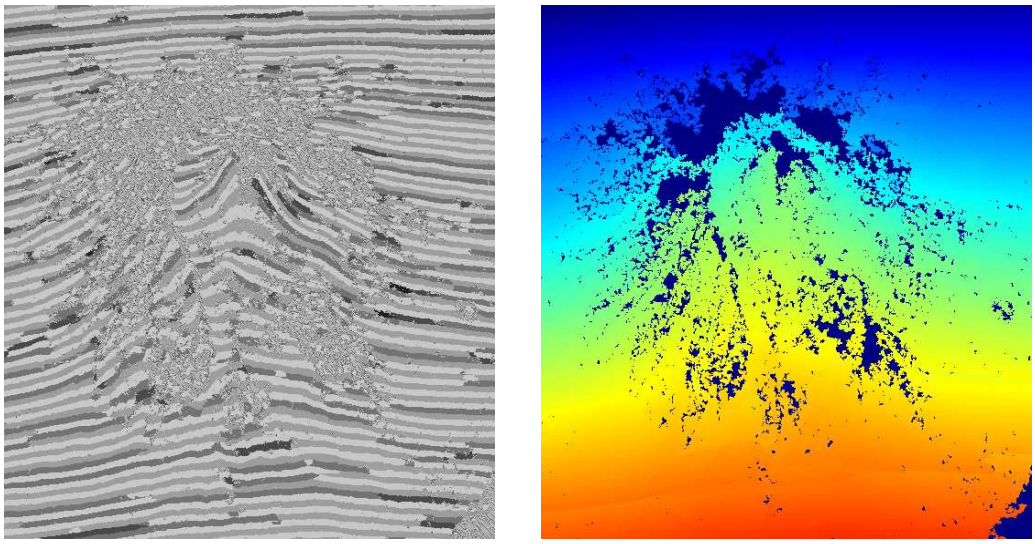
Segmentation result using zero residual [17] Measured digital elevation model

Figure 8.9: Results for the Mt. Vesuvius image using the probability of zero residual. Some unwrapping was not correctly done in the upper left hand corner in a "blue" colored transition. However, overall the segmentation and unwrapped phase are credible. The holes in the DEM can be filled in using interpolation [17]. The SNA-GM-ICM algorithm was run using schedule $\beta = \{0.5, 0.55, \ldots, 1.0, 1.05\}$ with on average 40 levels and 9 million site visits.

that is much more prevalent with a higher β.

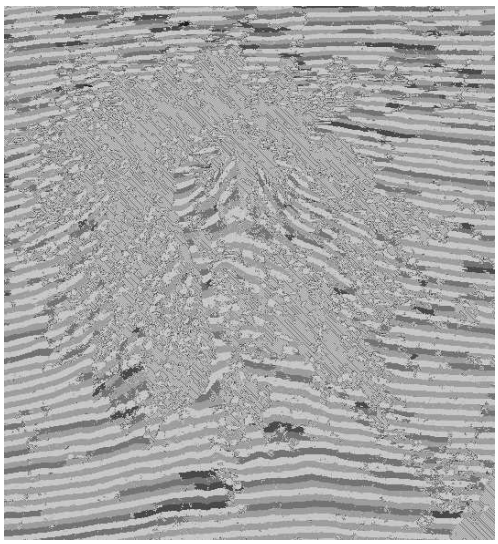
8.6 Conclusions

In this chapter, we have demonstrated that a principled, probability-based approach to phase unwrapping is feasible without a high computational cost. The proposed approach presents several advantages over other phase unwrapping algorithms. First, it is hierarchical and as such is able to use efficiently multiscale processing producing results in $O(N)$ time. Second, MRF models such as the edge Potts model are a natural choice for modelling measured phase since measured phase through its sharp discontinuities at multiples of 2π fits a piecewise constant function. Third, the use of a probabilistic distance measure (or

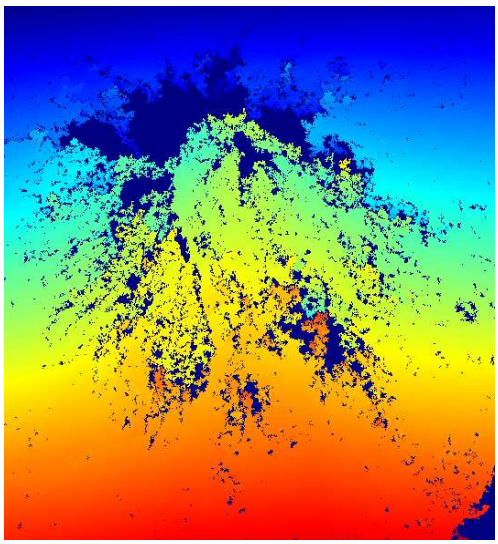


Segmentation result using model (8.5) Measured digital elevation model

Figure 8.10: Results for the Mt. Vesuvius image using (8.5): $w = 2$ and $\upsilon = 0.08$, $\beta = 0.6$. Very few problems (compared to Figure 8.11 thanks to a more aggressive pixel/region merging parameter β. However, problems start appearing in other areas of the reconstructed image (see especially reconstructed phases in the middle of the bottom of the image where due to region spilling phases are reconstructed incorrectly).

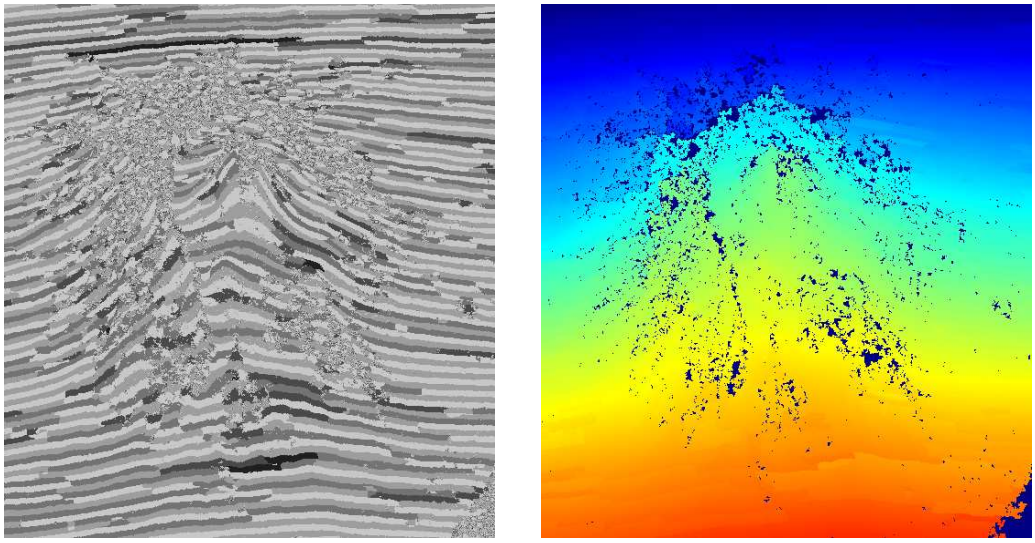


Segmentation result using model (8.5)



Measured digital elevation model

Figure 8.11: Results for the Mt. Vesuvius image using (8.5): $w = 2$ and $\upsilon = 0.08$, $\beta = 0.4$. Many problem thought the reconstructed phase image ("islands" of differently colored regions). This is due to the segmentation result which shows that most of the mountain areas which have low coherence have not been segmented. The SNA-ICM algorithm was run using $\beta = 0.4$ with on average 15 levels and 3.4 million site visits.



Segmentation result using model (8.5) Measured digital elevation model

Figure 8.12: Results for the Mt. Vesuvius image using (8.5): $w = 2$ and $\upsilon = 0.08$, $\beta = 0.75$. β causes most pixels to aggregate and therefore few unassigned pixels remain. However, the unwrapping has many problems such as the middle bottom area of the image as well as parts of the top portion (transitions from light blue to deep blue).

its approximation) allows for making principled decisions to determine which phases are similar and which ones are not given the amount of phase difference *and* the confidence in the phase embodied by the coherence. Finally, the model (8.5) is not as accurate as some of the previously determined models [17]; however, it is much simpler to implement and produces credible results (*e.g.*, better performance than Carballo's model on the synthetic image). The shortcomings in the results compared to network flow methods could also be attributed to the image segmentation framework used in this thesis. That is, the model for segmentation is edge-based and dependent only on pixel-to-pixel gradients. Perhaps a model using several pixels on each side of an edge to characterize an edge might prevent much of the region spilling that occurs. Using nested aggregation on the network flow framework should produce similar results to [17] with a considerable reduction in computational complexity and is left as a future exercise.

Chapter 9

Conclusions

9.1 Summary

The main motivation behind this thesis was to contribute to the state of the art in pixel similarity and pixel grouping methods. This objective was achieved by devising stochastic nested aggregation, an original fine-to-coarse irregular hierarchy of segmentations (or graph partitions), and designing new color semi-metrics. In summary, we made the following thesis contributions:

1. The introduction of stochastic nested aggregation gives us an alternative option for Gibbs sampling acceleration from graph cuts. The new method has geometrical convergence to the stationary probability $p(\ell)$ with a computational complexity of $O(N)$ which is a considerable improvement over $O(N^3)$ (for simulated annealing). The speed-up is more significant when homogenous regions within an image are large thanks to the pyramidal structure of nested aggregation. In practical terms, Gibbs sampling can be sped up by a factor of 1000-10000 (or more) depending on the graph size, the size of the largest partition in the coarsest level graph, and the optimization algorithm used within the SNA framework. Furthermore, stochastic nested aggregation does not need a stopping criterion as it is minimizing an energy function with a unique optimum point.

2. Stochastic nested aggregation also improved on Iterated Conditional Modes (ICM)

[7], a local deterministic approach to optimization, by breaking label configuration deadlocks which give rise to local minima. The creation of a new reduced order graph at each hierarchy level enables escaping from the local minimum as the structure of the grid surrounding the deadlocked region has been altered. Additionally, the speed-up for ICM was even more significant than for SA, enabling ICM to become a fast global deterministic optimization approach that is a competitive alternative to other global approaches.

3. The introduction of a Graduated Models strategy for Stochastic Nested Aggregation in order to avoid getting stuck in an undesirable local minimum (*e.g.*, avoid region-to-region spilling in image segmentation) is an important improvement on the single model. We applied Graduated Models successfully to the Potts energy model where we varied β, the region coupling parameter, from a low value (all pixels or nodes are their own regions) to the desired value (where regions homogenous in features have formed). Therefore, through careful nested aggregation simulated annealing and ICM converge to a very good local minimum. In the limit, when all edge weights are used in sequence, SNA-GM becomes the special case Highest Confidence First algorithm.

4. The stochastic nested aggregation framework allows us to use different models at various levels in the nested hierarchy. Thus, we introduced a region mean-based Potts energy which uses a region mean (as opposed to the pixel gradient-based computation in the classic Potts model) in order to compute pixel-to-region and region-to-region distances. We used the first principal component of the covariance matrix of a region's pixels (essentially the mean direction of the pixels) to represent this mean. However, the mean model proved to be an inadequate tool by itself. Therefore, we allowed SNA to first aggregate pixels into a region using the edge-based Potts model and carry out processing at coarser scales using the mean Potts model. Such a progression invalidates the condition that we are solving the same problem at all levels in the hierarchy which makes this process very difficult to analyze theoretically. However, practical image segmentation results showed the importance of this adaptation.

5. The Mixture of Principal Components (MPC) paradigm [34] where regions are de-

fined by the principal component vector corresponding to the largest eigenvalue of the covariance matrix of the data in each region was adapted to the Markov Random Field framework. Furthermore, the class or region prototypes were determined probabilistically by continuous Gibbs sampling from a region prototype distribution.

In the domain pixel distance measures, two problems were specifically of interest: physics-based color image segmentation of real world color scenes and phase unwrapping of interferometric Synthetic Aperture Radar (inSAR) images based on image segmentation. With respect to physics- or reflectance-based color distance measures, several contributions have been made:

1. We showed that projecting RGB pixels into a 2-dimensional subspace results in a highlight invariant color space in which a modified vector angle distance measure can be used to achieve shading invariance thus allowing for reflectance-based image segmentation. However, this type of processing does not take noise into account and is ambiguous for the zero vector where many saturated highlight pixels are projected.

2. Due to the unpredictable behavior of the vector angle distance measure for pixels with low RGB intensities, three new color distance measures were introduced. These distance measures are based on a probabilistic interpretation of color in order to create a shading invariant and noise resistant color distance measures in RGB: the Same Class Hypothesis distance measure, the Common Mean Hypothesis distance measure with most likely mean, and the Common Mean Hypothesis distance measure with equally likely mean.

3. The highlight invariant transformation was applied to the Same Class Hypothesis distance measure in order to create a new probabilistic distance measure that is both shading and highlight invariant.

4. Since the vector angle distance measure shows unpredictable behavior when pixel values have very low intensities, a vector angle accuracy criterion that trusts pixel values with high intensity and distrusts pixel values of low intensity was introduced for a Markov Random Field clustering-based application.

Finally, in the domain of pixel similarity for phase unwrapping problems, the application of stochastic nested aggregation with an edge Potts model to the interferometric synthetic aperture radar phase unwrapping problem using both coherence and phase information was done for the first time in this thesis. Furthermore, a new measure for carrying out the segmentation based on phase and coherence maps was devised and tested. This measure is an approximation of the model-based probabilistic cost function developed in [17].

9.2 Future Extensions

Many possible research avenues may be followed to improve on this work.

9.2.1 Stochastic Nested Aggregation

There are several unanswered questions that remain for the Stochastic Nested Aggregation framework:

1. A Graduated Models strategy was presented as a means to avoid local minima. The nature of local minima for the Potts model in graph partition in general and image segmentation in particular is that they are dependent both on the value of β and the structure of the graph itself. It would be useful to know what are the limitations on a β schedule and whether an optimal schedule can be derived.

2. A related idea is the relationship between edge gaps and the β schedule. As gaps in edges widen, the edge at that particular location becomes ambiguous. One useful question would be to ask at what point does an edge gap cease being a gap and become a narrow part of some region? Since this is a subjective topic, psychological experiments would be needed to assess the human point of view.

3. Many models of pixel and region interaction exist over and beyond the simple edge and mean Potts models applied in this thesis. For example, a region interaction model might be based on an edge between sets of pixels two to three rows wide. Such an edge might be much more robust against region-to-region spilling as many spills occur due to single pixel transitions between large regions. One could also

envision another mixed model where small regions are built using the edge model and the coarser scales are processed using a more complicated model.

4. Another type of model altogether could be applied to the nested aggregation framework. For example, regions could be represented by a prototype with distances computed between the region prototype and individual pixels or pixels within regions adjacent to that region. This model would function in much the same way like Markov Random Field-based clustering approaches [40, 107] and would minimize the total between class variation [150]. The advantage of this method would be to combine a classical global prototype-based method with a method which works in a local context (*i.e.*, an edge-based Potts model in the Markov Random Field framework).

5. Gibbs sampling is a very robust method to obtain samples from unknown distributions. However, other means exist such as the Metropolis-Hastings and the Generalized Metropolis-Hastings (of which Gibbs sampling and Metropolis-Hastings sampling are special cases) algorithms [88, 157]. Metropolis-Hastings could prove to be a much faster alternative given that only one test is performed for each node-to-node comparison rather than K tests based on K labels. This test can be rejected many times if it does not minimize the energy (or if a higher energy state is not accepted). The Gibbs sampler, on the other hand, eliminates the need for rejections since it tests all possible alternatives. In some practical applications, this difference in speed (at the possible cost of solution quality) could be a needed compromise. Furthermore, it is not certain that results would be much worse than those generated by the Gibbs sampler (they might still be acceptable) which is a proposition that must be tested.

6. One of the major limitation in showing the practicality of the nested aggregation framework has been the integration of texture discrimination into the Potts model. Many researchers have worked on this in the past (see [22, 88] for details of many methods) and many texture descriptors exist and can be integrated easily into this framework. Simple texture descriptors need to be tested to make sure that they can be used in the SNA framework.

7. There are several constraints that can be placed on a region other than a region or

pixel coupling term. They include the size of the region, shape and other attributes that might be important in a given segmentation task [94, 142]. These parameters have often a considerable impact on the functioning of a model such as Potts and their integration into a nested framework is not evident. Studies should be carried out integrating these parameters in an non-ad-hoc manner and testing their usefulness in generating meaningful segmentations.

8. Finally, the SNA framework could be integrated with others in order to devise a more robust graph partitioning framework. For example, integrating SNA with graph cuts could be very useful. Graph-cuts works by recursively subdividing an image until the desired energy is minimized while nested aggregation does the same from the bottom up. Both methods could be used to optimize the same function. One could take several of the levels for both methods prior to reaching the optimal solution and use an energy minimization scheme to optimize the final partitioning based on both results thus creating an "irregular" multi-grid monte carlo method. There would be difficulties associated with devising an appropriate energy-based solution since there would not be in practice a one-to-one correspondence between regions generated with graph cuts and nested aggregation (although in theory when minimizing the same global criterion the solution should the same).

9.2.2 Distance Measures

Probabilistic distance measures have been derived for computing shading and highlight invariant-based distances between color pixels to achieve color constancy in image segmentation. Much work remains in terms of the integration of the new distance semi-metrics with image segmentation algorithms.

1. There is a need to derive and test the highlight invariant Common Mean Hypothesis Test Distance Measure. The components of the "mean" vector in this method are necessarily dependent on each other since the covariance matrix is not diagonal (especially due to the highlight transformation). The computation of these means will have to be done using an iterative gradient descent method which is going to significantly impact the computational time of the segmentation algorithm. It would

be interesting to find out if the performance of this metric is better than the HI-SCH semi-metric (6.40).

2. One major limitation of the current approach is the specification of a single noise standard deviation or variance value. This is a very crude assumption that might be seriously flawed and perhaps could explain some of the pitfalls of current distance measure implementations. The optimal way to specify the variance of the noise would be to specify a different value for each intensity/color pair. This is not feasible using a single image and therefore a database of noise variances needs to be used [90].

3. Prototypes need to be developed for the new distance measures. The prototypes could be based on the formulas for the "means" (6.30) or (6.27) since we need to take into account variance when computing prototypes for image regions. These prototypes would allow the design of new mean-based Potts models that could more effectively segment images with shading and highlights when used at the higher levels of the SNA hierarchy.

Phase unwrapping is a vast field of research where much development has happened due to the practical importance of processing inSAR images. Many challenges remain in devising better cost functions for the segmentation of phase images. These cost functions or distance measures could also be used in other algorithms such as network flow. Given that the same cost functions were used in network flow [16] as in image segmentation in this thesis, a detailed analysis is needed to study why network flow performs better on synthetic images (4% error for network flow vs. 12% for stochastic nested aggregation using the edge Potts model). Furthermore a detailed study testing a larger range of parameters for the new model (8.5) is required (*e.g.*, fractional exponents could approximate better the shape of the zero residual function from [16]).

Appendix A

Hierarchical Model Equivalence Proof

Suppose we have a set of nodes $\mathcal{L}$; then, we have some energy function:

$$U = \sum_{i,j \in \mathcal{L}} \{\Phi_{i,j}\delta_{l_i,l_j} + \beta_{i,j}(1 - \delta_{l_i,l_j})\} \tag{A.1}$$

where l_i is the label of node i. Suppose we divide $\mathcal{L} = A \cup B$ such that *all* the nodes in B are grouped into one; then,

$$\begin{aligned} \bar{U} &= \textstyle\sum_{i,j \in A}\{\bar{\Phi}_{i,j}\delta_{l_i,l_j} + \bar{\beta}_{i,j}(1 - \delta_{l_i,l_j})\} + \\ &\quad \textstyle\sum_{i \in A, j=b}\{\bar{\Phi}_{i,j}\delta_{l_i,l_j} + \bar{\beta}_{i,j}(1 - \delta_{l_i,l_j})\} \end{aligned} \tag{A.2}$$

and where b is the node index corresponding to B. Therefore,

$$\begin{aligned} U - \bar{U} &= \textstyle\sum_{i,j \in A}\{(\Phi_{i,j} - \bar{\Phi}_{i,j})\delta_{l_i,l_j} + (\beta_{i,j} - \bar{\beta}_{i,j})(1 - \delta_{l_i,l_j})\} + \\ &\quad \textstyle\sum_{i \in A, j \in B}\{\Phi_{i,j}\delta_{l_i,l_j} + \beta_{i,j}(1 - \delta_{l_i,l_j})\} - \\ &\quad \textstyle\sum_{i \in A, j=b}\{\bar{\Phi}_{i,j}\delta_{l_i,l_j} + \bar{\beta}_{i,j}(1 - \delta_{l_i,l_j})\} + \\ &\quad \textstyle\sum_{i,j \in B}\{\Phi_{i,j}\delta_{l_i,l_j} + \beta_{i,j}(1 - \delta_{l_i,l_j})\} - 0 \end{aligned} \tag{A.3}$$

We can simplify this equation by setting $\bar{\beta}_{i,j} = \beta_{i,j}$ and $\bar{\Phi}_{i,j} = \Phi_{i,j}$ for $\forall i, j \in A$. Furthermore, since by definition the region with label l_b corresponds to all regions with all l_j $\forall j \in B$; therefore, we can assume that $l_b = l_j$ $\forall j \in B$ and by extension $\bar{\beta}_{i,b} = \sum_{j \in B} \beta_{i,j}$

and $\bar{\Phi}_{i,b} = \sum_{j \in B} \Phi_{i,j}$. This then leaves

$$U - \bar{U} = \sum_{i,j \in B} \{\Phi_{i,j}\delta_{l_i,l_j} + \beta_{i,j}(1 - \delta_{l_i,l_j})\} \tag{A.4}$$

which further simplifies to

$$U - \bar{U} = \sum_{i,j \in B} \Phi_{i,j} \tag{A.5}$$

since $\delta_{l_i,l_j} = 1 \ \forall i, j \in B$. Therefore, under the proposed grouping the energy function U and $\bar{U}$ differ only by a constant. Finally, if the optimum solution to U satisfies $l_i = l_j$ $\forall i, j \in B$, then by definition U and $\bar{U}$ have the same optimum. If $l_i \neq l_j$ for some $i, j \in B$, then the equations do not hold and the equivalency cannot be established.

Bibliography

[1] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 641-647, 1994.

[2] J. Angulo, and J. Serra, "Color segmentation by ordered mergings," *IEEE International Conference on Image Processing*, vol. 2, pp. 125-128, 2003.

[3] A. Barbu and S.C. Zhu, "Multigrid and multi-level Swendsen-Wang cuts for hierarchic graph partition," *IEEE Computer Vision and Pattern Recognition Conference*, pp. 731-738, 2004.

[4] A. Barbu and S.C. Zhu, "Graph Partition by Swendsen-Wang Cut," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1239-1253, 2005.

[5] J. Batille, A. Casals, J. Freixenet, and J. Marti, "Review on strategies for recognizing natural objects in colour images of outdoor scenes," *Image and Vision Computing*, vol. 18, no. 6, pp. 515-530, May 2000.

[6] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *Journal of the Royal Statistical Society B*, vol. 36, pp. 192-236, 1974.

[7] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society B*, vol. 48, pp. 259302, 1986.

[8] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*, Plenum, New York, 1981.

[9] A. Blake, and A. Zisserman, *Visual Reconstruction*, MIT Press, Cambridge, MA, 1987.

[10] A. Bovik (Editor). *Handbook of Image and Video Processing*, Academic Press, San Diego, CA, 2000.

[11] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222-1239, 2001.

[12] L. Brun, and W. Kropatsch, "Combinatorial pyramids," *IEEE International Conference on Image Processing*, vol. 2, pp. 33-36, Sept. 2003.

[13] L. Brun, M. Mokhtari, and F. Meyer, "Hierarchical watersheds within the Combinatorial Pyramid framework," *International Conference Discrete Geometry for Computer Imagery*, pp. 34-44, 2005.

[14] S. D. Buluswar, and B. A. Draper, "Color Models for Outdoor Machine Vision," *Computer Vision and Image Understanding*, vol. 85, no. 2, pp. 71-99, February 2002.

[15] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, November 1986.

[16] G. Carballo, *Statistically-Based Multiresolution Network Flow Phase Unwrapping for SAR Interferometry*, Ph.D. Thesis, University of Waterloo, 2000.

[17] G.F. Carballo and P.W. Fieguth, "Hierarchical Network Flow Phase Unwrapping," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 8, August 2002.

[18] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: image segmentation using expectation-maximization and its application to image querying," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1026-1038, Aug. 2002.

[19] V. Cerny, "A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, pp. 41-51, 1985.

[20] M. M. Chang, M. I. Sezan, and A. M. Tekalp, "Adaptive Bayesian segmentation of color images," *Journal of Electronic Imaging*, vol 3, no. 4, pp.404-414, October 1994.

[21] R. Chellappa, "Two-dimensional discrete gaussian Markov random field models for image processing," in L. N. Kanal and A. Rosenfeld (Eds.), *Progress in Pattern Recognition 2*, pp. 79-112, 1985.

[22] R. Chellappa, and A. Jain, Editors, *Markov Random Fields: Theory and Application*, Academic Press, New York, 1993.

[23] C. W. Chen and H. A. Zebker, "Network approaches to two-dimensional phase unwrapping: intractability and two new algorithms," *Journal of the Optical Society of America A*, vol. 17, pp. 401-414, 2000.

[24] H. D. Cheng, , X. H. Jiang, Y. Sun and J. Wang, "Color image segmentation: advances and prospects," *Pattern Recognition*, vol. 34, no. 12, pp. 2259-2281, December 2001.

[25] P. B. Chou, and C. M. Brown, "The theory and practice of Bayesian image labelling," *International Journal of Computer Vision*, vol. 4, pp. 185-210, 1990.

[26] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5 , pp. 603-619, May 2002.

[27] D. Comaniciu, "Image segmentation using clustering with saddle point detection," *IEEE International Conference on Image Processing*, vol. 3, pp. 297-300, 2002.

[28] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, second edition, MIT Press and McGraw-Hill, 2001.

[29] M. Costantini, "A phase unwrapping method based on network programming," *1996 Fringe Workshop ERS SAR Interferometry*, ESA SP-406, pp. 261-272, 1996.

[30] M. Costantini, "A novel phase unwrapping method based on Network Programming," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, pp. 813-821, 1998.

[31] M. J. Daily, "Color Image Segmentation Using Markov Random Fields," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 304-312, 1989.

[32] Y. Deng, and B. S. Manjunath, "Unsupervised Segmentation of Color-Texture Regions in Images and Video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 800-810, August 2001.

[33] M. Denney, "Introduction to importance sampling in rare-event simulations," *European Journal of Physics*, vol. 22, pp. 403-411, 2001.

[34] R. D. Dony, and S. Haykin, "Image segmentation using a mixture of principal components representation," *IEE Vision, Image & Signal Processing*, vol. 144, pp. 73-80, April 1997.

[35] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed., John Wiley & Sons, New York, USA, 2001.

[36] M. Eineder, M. Hubig, B. Milcke, "Unwrapping large interferograms using the minimum cost flow algorithm," *IEEE International Geoscience and Remote Sensing Symposium*, pp. 83-87, 1998.

[37] K.-B. Eum, J. Lee, and A. N. Venetsanopoulos, "Color image segmentation using a possibilistic approach," *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, pp. 1150-1155, Beijing, China, 1996.

[38] J. Fan, D.K.Y. Yau, A.K. Elmagarmid and W.G. Aref, "Automatic image segmentation by integrating color-edge extraction and seeded region growing," *IEEE Transactions on Image Processing*, vol. 10, no. 10 , October 2001, pp. 1454-1466.

[39] P. Felzenszwalb and D. P. Huttenlocher, "Efficient Graph-Based Image Segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167-181, 2004.

[40] P. Fieguth and S. Wesolkowski, "Highlight and Shading Invariant Color Image Segmentation Using Simulated Annealing," *Energy Minimization Methods in Computer Vision and Pattern Recognition*, Sophia-Antipolis, France, pp. 314-327, September 2001.

[41] P. Fieguth, *Statistical Processing of Multidimensional Data*, Systems Design Engineering, University of Waterloo, 2006. (unpublished manuscript)

[42] P. Fieguth, Private Communication, 2007.

[43] G Fornaro, G Franceschetti, R Lanari, D Rossi, and M. Tesauro M, "Interferometric SAR Phase Unwrapping via Finite Elements Method," *IEE Radar, Sonar, Navigation*, vol. 144, no. 5, pp. 266-274, 1997.

[44] K. S. Fu, and J. K. Mui, "A survey of image segmentation," *Pattern Recognition*, vol.13, pp. 3-16, 1981.

[45] H. Gao, W.-C. Siu, and C.-H. Hou, "Improved techniques for automatic image segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 12, pp. 1273-1280, December 2001.

[46] J. Gao, J. Zhang, and M. G. Fleming, "Novel technique for multiresolution color image segmentation," *Optical Engineering*, vol. 41, no. 3, pp. 608-614, 2002.

[47] Y. Gdalyahu, D. Weinshall, and M. Werman, "Self-organization in vision: Stochastic clustering for image segmentation, perceptual grouping, and image database organization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1053-1074, October 2001.

[48] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, 1984.

[49] T. Gevers and A.W.M. Smeulders, "Color-based object recognition," *Pattern Recognition*, vol. 32, pp. 453-464, 1999.

[50] T. Gevers, "Adaptive Image Segmentation by Combining Photometric Invariant Region and Edge Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, June 2002, pp. 848-852.

[51] D.C. Ghiglia and L.A. Romero, "Robust two-dimensional weighted and unweighed phase unwrapping that uses fast transforms and iterative methods," *Journal of the Optical Society of America A*, vol. 11, no. 1, pp. 107117, 1994.

[52] D. Ghiglia and M. Pritt, *Two-Dimensional Phase Unwrapping. Theory, Algorithms, and Software*, John Wiley & Sons, New York, 1998.

[53] R.M. Goldstein, H.A. Zebker, and C.L. Werner, "Satellite radar interferometry: Two dimensional phase unwrapping," *Radio Science*, vol. 23, no. 4, pp. 717-720, August 1988.

[54] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Addison-Wesley, 1993.

[55] J. Goodman and A.D. Sokal, "Multigrid Monte Carlo method. Conceptual foundations," *Physics Review D*, vol. 40, no. 6, pp. 2035-2071, September 1989.

[56] A. Gray and P. Farris-Manning, "Repeat-pass interferometry with airborne synthetic aperture radar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 31, pp. 180191, 1993.

[57] T. Guyennem and D. Danesy (Editors), *1996 Fringe Workshop ERS SAR Interferometry*, ESA SP-406, 1996. (http://www.geo.unizh.ch/rsl/fringe96/)

[58] R. M. Haralick and L. G. Shapiro, "Survey: Image segmentation techniques," *Computer Visual & Graphical Image Processing*, vol. 29, pp. 100-132, 1985.

[59] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. 1, Addison-Welsey, Reading, MA, 1992.

[60] G. G. Hazel, "Multivariate Gaussian MRF for multispectral scene segmentation and anomaly detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 3, pp. 1199 -1211, May 2000.

[61] G. Healey, and T. O. Binford, "A color metric for computer vision," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10-17, Ann Arbor, 1988.

[62] G. Healey, "Segmenting Images Using Normalized Color," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 1, pp. 64-73, 1992.

[63] G. E. Healey and R. Kondepudy, "Radiometric CCD camera calibration and noise estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 3, pp. 267276, 1994.

[64] M. Hedley and H. Yan, "Segmentation of color images using spatial and color space information," *Journal of Electronic Imaging*, vol. 1, pp. 374-380, October 1992.

[65] T. Hofmann and J. M. Buhmann. "Pairwise Data Clustering by Deterministic Annealing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 1, pp. 1-14, 1997.

[66] B. K. P. Horn, "Understanding image intensities," *Artificial Intelligence*, vol. 8, 1977, pp. 201-231.

[67] C. L. Huang, T. Y. Cheng, and C. C. Chen, "Color image segmentation using scale space filter and Markov random field," *Pattern Recognition*, vol. 25, no. 10, pp. 1217-1229, 1992.

[68] A. K. Jain, *Fundamentals of digital image processing*. Prentice Hall, Englewood Cliffs, NJ, 1989.

[69] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Prentice Hall, Englewood Cliffs, N.J. 1988.

[70] R.M. Karp, "Reducibility Among Combinatorial Problems," *Complexity of Computer Computations Symposium, IBM Thomas J. Watson Research Center*, pp. 85-103, Plenum Press, New York, 1972.

[71] M. Kass, A. Witkin, and D. Terzolpoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321331, 1988.

[72] Z. Kato, M. Berthod, and J. Zeroubia, "A Hierarchical Markov Random Field Model and Multitemperature Annealing for Parallel Image Classification," *Graphical Models and Image Processing*, vol. 58, no. 1, 1996, pp. 18-37.

[73] Z. Kato, T.C. Pong, J.M. Lee, "Color image segmentation and parameter estimation in a markovian framework," *Pattern Recognition Letters*, vol. 22, no. 3-4, pp. 309-321, March 2001.

[74] I. B. Kerfoot, and Y. Bresler, "Theoretical analysis of multispectral image segmentation criteria," *IEEE Transactions on Image Processing*, vol. 8, no. 6, pp. 798-820, June 1999.

[75] A. Khotanzad and o. J. Hernandez, "Color image retrieval using multispectral random field texture model and color content features," *Pattern Recognition*, vol. 36, no. 8, pp. 1679-1694, August 2003.

[76] S. Kirkpatrick, C. Gelatt, and M. Vecci, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671-680, 1983.

[77] G. J. Klinker, S. A. Shafer and T. Kanade, "A Physical Approach to Color Image Understanding," *International Journal of Computer Vision*, vol. 4, no. 1, pp. 7-38, 1990.

[78] G. J. Klinker, *A physical approach to color image understanding*. A.K. Peters, Wellesley, Mass., 1993.

[79] A. Koschan and M. Abidi, "Detection and classification of edges in color images," *IEEE Signal Processing Magazine*, vol. 22, pp. 64-73, January 2005.

[80] S. Kullback and R. A. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79-86, March 1951.

[81] F. Kurugöllü, and B. Sankur, "Color image segmentation using histogram multithresholding and fusion," *Image and Vision Computing*, vol. 19, no. 13, pp. 915-928, November 2001.

[82] S. Kuttikkad and R. Chellappa, "Statistical modeling and analysis of high-resolution Synthetic Aperture Radar images," *Statistics and Computing*, vol. 10, no. 2, pp. 133-145, 2000.

[83] J.-H. Lee, B.-H. Chang, and S.-D. Kim, "Comparison of Colour Transformations for Image Segmentation," *Electronics Letters*, vol. 30, no. 20, pp. 1660-1661, Sept. 1994.

[84] T. C. M. Lee, "A Minimum Description Length-Based Image Segmentation Procedure, and Its Comparison with a Cross-Validation-Based Segmentation Procedure," *Journal of the American Statistical Association*, vol. 95, no. 449, pp. 259-270, March 2000.

[85] O. Lezoray, and H. Cardot, "Cooperation of color pixel classification schemes and color watershed: A study for microscopic images," *IEEE Transactions on Image Processing*, vol. 11, no. 7, pp. 783-789, July 2002.

[86] C. T. Li, "Multiresolution image segmentation integrating Gibbs sampler and region merging algorithm," *Signal Processing*, vol. 83, no. 1, pp. 67-78, January 2003.

[87] S. Z. Li, "Modeling Image Analysis Problems Using Markov Random Fields," in C.R. Rao and D.N. Shanbhag (ed), *Stochastic Processes: Modeling and Simulation*, *vol. 20 of Handbook of Statistics*, Elsevier Science, pp. 1-43, 2000.

[88] S. Z. Li, *Markov Random Field Modelling in Image Analysis*, Springer, Tokyo, Japan, 2001.

[89] Y.W. Lim, and S.U. Lee, "On the color image segmentation algorithm based on the thresholding and fuzzy c-means techniques," *Pattern Recognition*, vol. 23, no. 9, pp. 1235-1252, 1990.

[90] C. Liu, W.T. Freeman, R. Szeliski and S.B. Kang. "Noise estimation from a single image," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 901-908, 2006.

[91] J. Liu and Y.-H. Yang, "Multiresolution color image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 689-700, July 1994.

[92] L. Lucchese and S. K. Mitra, "Color Segmentation Based on Separate Anisotropic Diffusion of Chromatic and Achromatic Channels," *IEE Proceedings Vision, Image, and Signal Processing*, vol. 148, no. 3, pp. 141-150, June 2001.

[93] L. Lucchese and S. K. Mitra, "Color Image Segmentation: A State-of-the-Art Survey," *Proceedings of the Indian National Science Academy (INSA-A)*, New Delhi, India, vol. 67, A, no. 2, pp. 207-221, March 2001.

[94] J. Luo and C. Guo, "Perceptual grouping of segmented regions in color images," *Pattern Recognition*, vol. 36, pp. 2781-2792, 2003.

[95] Q.-T. Luong, "Color in computer vision," in *Handbook of Pattern Recognition and Computer Vision*, C.H. Chen, L.F. Pau and P.S.P. Wang (Eds.), pp. 311-368, World Scientific, 1993.

[96] B. A. Maxwell and S. A. Shafer, "Physics-based segmentation of complex images using multiple hypotheses of image formation," *Computer Vision and Image Understanding*, vol. 65, no. 2, pp. 269-295, 1997.

[97] B. A. Maxwell and S. A. Shafer, "Segmentation and interpretation of multicolored objects with highlights," *Computer Vision and Image Understanding*, vol. 77, no. 1, pp. 1-24, January 2000.

[98] M. Mirmehdi and M. Petrou, "Segmentation of color textures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 2, pp. 142-159, 2000.

[99] A. Montanvert, P. Meer, and A. Rosenfeld, "Hierarchical image analysis using irregular tessellations," *IEEE Transactions on Pattern Analvsis and Machine Intelligence*, vol. 13, no. 4, pp. 307-316, April 1991.

[100] B.J. Moorman and P.W. Vachon, "Detecting ground ice melt with interferometric synthetic aperture radar," *20th Canadian Symposium on Remote Sensing*, Calgary, Alberta, May 10-13, 1998.

[101] J. Mukherjee, "MRF clustering for segmentation of color images," *Pattern Recognition Letters*, vol. 23, no. 8, pp. 917-929, 2002.

[102] S. G. Narasimhan, and S. K. Nayar, "Vision and the atmosphere," *International Journal of Computer Vision*, vol. 48, no. 3, pp. 233-254, 2002.

[103] S. K. Nayar, K. Ikeuchi, and T. Kanade, "Determining the shape and reflectance of hybrid surfaces by photometric sampling," *IEEE Transactions on Robotic Automation*, vol. 6, 1990, pp. 418-431.

[104] H. H. Nguyen, and P. Cohen, "Gibbs random fields, fuzzy clustering, and the unsupervised segmentation of textured images," *CVGIP: Graphical Models and Image Processing*, vol.55, no.1, pp.1-19, 1993.

[105] N. R. Pal and S.K. Pal, "A Review on Image Segmentation Techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277-1294, 1995.

[106] D. K. Panjwani, and G. Healey, "Markov Random Field Models for Unsupervised Segmentation of Textured Color Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 10, 1995.

[107] T. N. Pappas, "An adaptive clustering algorithm for image segmentation," *IEEE Transactions on Signal Processing*, vol. 40, no. 4, pp. 901-914, 1992.

[108] S. Parashar, E. Langham, and S. Ahmed, "RADARSAT-I system commissioning and beyond," *IEEE Canadian Conference on Electrical and Computer Engineering*, vol. 1, pp. 5-12, 1996.

[109] J. B. Park and A. C. Kak, "A Truncated Least Squares Approach to the Detection of Specular Highlights in Color Images," *IEEE International Conference on Robotics and Automation (ICRA 2003)*, 2003.

[110] B. T. Phong, "Illumination for computer generated images," *Communications of the ACM*, vol. 18, 1975, pp. 311-317.

[111] S. H. Park, I. D. Yun, and S.U. Lee, "Color Image Segmentation Based on 3-D Clustering: Morphological Approach," *Pattern Recognition*, vol. 31, no. 8, 1998, pp. 1061-1076.

[112] K. N. Plataniotis, A. N. Venetsanopoulos, *Color image processing and applications*. Springer: Berlin, 2000.

[113] M. Pritt, "Phase Unwrapping by means of Multigrid techniques for Interferometric SAR," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, no. 3, pp. 728-738, 1996.

[114] S. Ray, R. H. Turi and P. E. Tischer, "Clustering-based colour image segmentation: An evaluation study," *Proceedings Digital Image Computing: Techniques and Applications Conference*, Brisbane, Australia, pp. 86-92, 1995.

[115] S. Ray and R. H. Turi, "Determination of number of clusters in K-means clustering and application in colour image segmentation," in N. R. Pal, A. K. De and J. Das (Eds.), *Proceedings of the International Conference on Advances in Pattern Recognition and Digital Techniques (ICAPRDT)*, pp. 137-143, Calcutta, India, 1999.

[116] N. Robertson, D.P. Sanders, P.D. Seymour and R. Thomas, "The four colour theorem," *Journal Combinatorics Theory Series B*, vol. 70, pp. 2-44, 1997.

[117] P.A. Rosen, S. Hensley, I.R. Joughin, F.K. Li, S.N. Madsen, E. Rodrguez, and R.M. Goldstein, "Synthetic aperture radar interferometry," *Proceedings of the IEEE*, vol. 88, no. 3, pp. 333-382, 2000.

[118] S. Roy, and I. Cox, "A maximum-flow formulation of the n-camera stereo correspondence problem," *IEEE International Conference on Computer Vision*, Bombay, India, 1998.

[119] E. Saber, A.M. Tekalp, and G. Bozdagi,"Fusion of Color and Edge Information for Improved Segmentation and Edge Linking," *Image and Vision Computing*, vol. 15, 1997.

[120] R. Schettini, "A segmentation algorithm for color images," *Pattern Recognition Letters*, vol. 14, pp. 499-506, June 1993.

[121] S. Sclaroff and L. Liu, "Deformable Shape Detection and Description via Model-Based Region Grouping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 5, pp. 475-489, 2001.

[122] L. Shafarenko, M. Petrou, and J. Kittler, "Automatic watershed segmentation of randomly textured color images," *IEEE Transactions on Image Processing*, vol. 6, pp. 1530-1544, November 1997.

[123] S. A. Shafer, "Using Color to Separate Reflection Components," *Color Research and Application*, vol. 10, no. 4, pp. 210-218, 1985.

[124] J. Shi, and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, 2000.

[125] H. Stokman and T. Gevers, "Photometric Invariant Region Detectioonin Multi-Spectral Images," *Proceedings of Vision Interface*, pp. 90-96, Trois-Rivihres, Canada, 1999.

[126] J. Strand, T. Taxt, A. K. Jain, "Two-dimensional phase unwrapping using a block least-squares method," *IEEE Transactions on Image Processing*, vol. 8, no. 3, pp. 375-386, March 1999.

[127] K.-K. Sung, "A Vector Signal Processing Approach to Color", Technical report AITR-1349, AI Lab, MIT, Jan 1992.

[128] R.H. Swendsen and J.S. Wang, "Nonuniversal critical dynamics in Monte Carlo simulations," *Physical Review Letters*, vol.58, no. 2, pp.86-88, 1987.

[129] A.J.P. Theuwissen, *Solid-State Imaging with Charge-Coupled Devices,* Kluwer: Dordrecht, Netherlands, 1995.

[130] B. Thirion, B. Bascle, V. Ramesh, and N. Navab, "Fusion of Color, Shading and Boundary Information for Factory Pipe Segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 349-356, Hilton Head, USA, 2000.

[131] D. Titterington, A. Smith, and U. Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley: New York, 1985.

[132] S. Tominaga and B. Wandell, "The standard reflectance model and illuminant estimation", *Journal of Optical Society of America A*, vol. 6, no.4, pp. 576-584, April 1989.

[133] S. Tominaga, "Surface Identification Using the Dichromatic Reflection Model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 658-670, July 1991.

[134] S. Tominaga, "Color Classification of Natural Color Images," *Color Research and Application*, vol. 17, no. 4, pp. 230-239, 1992.

[135] S. Tominaga, "Dichromatic Reflection Models for a Variety of Materials," *Color Research and Application*, vol. 19, no. 4, pp.277 - 285, 1994.

[136] S. Tominaga, "Spectral imaging by a multichannel camera," *Journal of Electronic Imaging*, vol. 8, no. 4, pp. 332-342, 1999.

[137] A. Tremeau, and N. Borel, "A Region Growing and Merging Algorithm to Color Segmentation," *Pattern Recognition*, vol. 30, no. 7, pp. 1191-1203, 1997.

[138] A. Trémeau, and P. Colantoni, "Regions adjacency graph applied to color image segmentation," *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 735-744, April 2000.

[139] Y. Tsin, V. Ramesh, and T. Kanade, "Statistical calibration of CCD imaging process," *IEEE International Conference on Computer Vision*, pp. 480487, 2001.

[140] Z. Tu, and S.-C. Zhu, "Image segmentation by data-driven Markov Chain Monte Carlo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 657-673, May 2002.

[141] I. Vanhamel, I. Pratikakis, and H. Sahli, "Multiscale gradient watersheds of color images," *IEEE Transactions on Image Processing*, vol. 12, no. 6, pp. 617-626, June 2003.

[142] O. Veksler, "Image Segmentation by Nested Cuts," *IEEE Computer Vision and Pattern Recognition Conference*, pp.339-344, June 2000.

[143] L. A. Vese, and T. F. Chan, "A multiphase level set framework for image segmentation using the Mumford and Shah model," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 271-293, December 2002.

[144] W. T. Vetterling, W. H. Press, B. P. Flannery, and S. A. Teukolsky, *Numerical Recipes in C++: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 2002.

[145] B.A. Wandell. *Foundations of Vision*, Sinauer Associates, Inc. Publishers, Sunderland, MA, 1995.

[146] W. Wang, C. Sun, and H. Chao, "Color Image Segmentation and Understanding through Connected Components," *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, pp. 1089-1093, October 1997.

[147] Y. Wang and P. Bhattacharya, "On Parameter-Dependent Connected Components of Gray Images," *Pattern Recognition*, vol. 29, pp. 1359-1368, 1996.

[148] S. Wesolkowski, M.E. Jernigan, R.D. Dony, "Global Color Image Segmentation Strategies: Euclidean Distance vs. Vector Angle," in Y.-H. Hu, J. Larsen, E. Wilson and S. Douglas (Eds.), *Neural Networks for Signal Processing IX*, IEEE Press, Piscataway, NJ, 1999, pp. 419-428.

[149] S. Wesolkowski, *Color Image Edge Detection and Segmentation: A Comparison of the Vector Angle and the Euclidean Distance Color Similarity Measures*, Master's thesis, Systems Design Engineering, University of Waterloo, Canada, 1999.

[150] S. Wesolkowski, S. Tominaga, and R.D. Dony, "Shading and Highlight Invariant Color Image Segmentation Using the MPC Algorithm," *SPIE Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts VI*, San Jose, USA, January 2001, pp. 229-240.

[151] S. Wesolkowski, and P. Fieguth. "Color Image Segmentation Using a Region Growing Method," *Ninth Congress of the International Colour Association*, Rochester, NY, June 2001.

[152] S. Wesolkowski and P. Fieguth, "Adaptive Color Image Segmentation Using Markov Random Fields," *IEEE International Conference on Image Processing 2002*, vol. III, pp. 769-772. Rochester, NY, September 2002.

[153] S. Wesolkowski and M. E. Jernigan, "Intensity-Invariant Color Image Segmentation Using MPC Algorithm," *IEEE/INNS International Joint Conference on Neural Networks*, pp. 200-205, Portland, USA, 2003.

[154] S. Wesolkowski, and P. Fieguth, "Color Image Segmentation using Connected Regions," *IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 1203-1206, 2003.

[155] S. Wesolkowski and P. Fieguth, "Hierarchical Region-Based Gibbs Random Field Image Segmentation," *International Conference on Image Analysis and Recognition*, Porto, Portugal, Sept 2004.

[156] S. Wesolkowski and P. Fieguth, "Hierarchical Region Mean Based Image Segmentation," *Canadian Conference on Computer and Robot Vision*, Quebec City, Canada, June 2006.

[157] G. Winkler, *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*, Springer-Verlag, Berlin, Germany, 1995.

[158] W. A. Wright, "Markov random field approach to data fusion and color segmentation," *Image and Vision Computing*, vol. 7, pp. 144-150, 1989.

[159] J. Wu, H. Yan, and A. N. Chalmers, "Color image segmentation using fuzzy clustering and supervised learning," *Journal of Electronic Imaging*, vol. 3, no. 4, pp. 397-403, October 1994.

[160] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: theory and application to image segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1101-1113, 1993.

[161] G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, Wiley: NYC, 1982.

[162] W. Xu and I. Cumming, "A region growing algorithm for InSAR phase unwrapping," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 1, pp. 124-134, 1999.

[163] T. Yamazaki and D. Gingras, "Unsupervised Multispectral Image Classification Using MRF Models and VQ Method," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 2, pp. 1173-1176, March 1999.

[164] J.S. Yedidia, W.T. Freeman, and Y. Weiss, "Generalized belief propagation," *MERL Report TR-2000-26*, 2000.

[165] J. Zhou, D. Lopresti, and T. Tasdizen, "Finding Text in Color Images," *Document Recognition V, Proc. SPIE vol. 3305*, pp. 130-140, 1998.

[166] S.C. Zhu and A. Yuille, "Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884 -900, Sept. 1996.

[167] S.-C. Zhu, "Statistical Modeling and Conceptualization of Visual Patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 691-712, 2003.

[168] D. Zugaj, and V. Lattuati, "A New Approach of Color Images Segmentation Based on Fusing Region and Edge Segmentations Outputs," *Pattern Recogniton*, vol. 31, no. 2, pp. 105-113, February 1998.